Diplomarbeit

FOM – Fachhochschule für Oekonomie & Management Essen

Angestrebter Abschluss
Diplom-Informatiker (FH)

7. Semester

Diplomarbeit zum Thema

Paket zur Erstellung wissenschaftlicher Arbeiten in LaTeX

Betreuer: Professor Dr.-Ing. Torsten Finke

Autor: AxEl Kesseler (134967)

Kaarst, den 28. Februar 2008

Veröffentlichungshinweis

Die vorliegende Diplomarbeit kann ohne Einschränkung der Öffentlichkeit zugänglich gemacht werden. Es wird sogar ausdrücklich darum gebeten. Jedoch ist jedwede kommerzielle Nutzung mit oder durch diese Abhandlung oder auch nur durch Teile dieser Abhandlung ausgeschlossen!

Inhalt

Vereinbarungen

package .... Paketbezeichner verwenden zur Darstellung die Schriftart Sans Serif.

`file` .... Datei- und Programmnamen verwenden zur Darstellung die Schriftart `Typewriter`.

`\macro` .... Makros, respektive Kommandos, werden mit einem umgekehrten Schrägstrich (backslash) eingeleitet und verwenden zur Darstellung die Schriftart `Typewriter`.

`environment` ..... Umgebungen verwenden zur Darstellung die Schriftart `Typewriter`.

`option` .... Optionen verwenden zur Darstellung die Schriftart `Typewriter`.

defval .... Standardwerte verwenden zur Darstellung die Schriftart *Italics*.

⟨*meta*⟩ .... Metadaten, also Platzhalter für Informationen jeglicher Art, verwenden zur Darstellung die Schriftart *Italics* und sind zusätzlich noch in spitzen Klammern eingeschlossen.

`{`⟨*mustarg*⟩`}` .... Pflichtparameter werden in geschweiften Klammern eingeschlossen und verwenden zur Formatierung der Daten die Darstellung ⟨*meta*⟩.

`[`⟨*optarg*⟩`]` .... Optionale Parameter werden in eckigen Klammern eingeschlossen und verwenden zur Formatierung der Daten die Darstellung ⟨*meta*⟩.

Abkürzungen

ASCII .... American Standard Code for Information Interchange

DIN .... Deutsches Institut für Normierung

EN .... Europäische Normen

HTML .... Hypertext Markup Language

ISO .... Internationale Organisation für Normung

PDF .... Portable Document Format

WYSIWYG .... What you see is what you get

Abbildungen

Quelltextbeispiele

Tabellen

1 Einleitung

Ziel dieser Diplomarbeit ist es, allen Studierenden ein Werkzug bereitzustellen, mit dessen Hilfe die Anfertigung der im Rahmen des Studiums zu absolvierenden wissenschaftlichen Abhandlungen erleichtert werden kann. Nur sollten nicht ausschließlich Studenten von den Ergebnissen der hier vorliegenden Arbeit profitieren. Deshalb richtet sich diese Diplomarbeit an all jene, die ihren Lesern die „Früchte ihrer Arbeit“ in einer ansprechenden Form darbieten möchten.

Sicherlich sind zur Verbesserung des Verständnisses der im Verlauf dieser Abhandlung dargebotenen Inhalte einige Grundkenntnisse im Umgang mit dem Textsatzsystem LaTeX[1] empfehlenswert und auch hilfreich, doch nicht unbedingt erforderlich. Dennoch soll diese Arbeit nicht als „Einführung in LaTeX“ missverstanden werden, da sie im Kern die speziellen Anforderungen der Erstellung wissenschaftlicher Abhandlungen fokussiert.

Nachdem zuvor noch einige grundlegende Gesichtspunkte behandelt werden, wird sich Kapitel 2 mit der grundsätzlichen Betrachtung wichtiger Elemente in wissenschaftlichen Abhandlungen befassen. Im daran anschließenden Kapitel 3 sollen dann die Ergebnisse der Umsetzung des *Pakets zur Erstellung wissenschaftlicher Arbeiten* näher untersucht werden. Hieran anschließend wird das Resümee in Kapitel 4 alle Betrachtungen dieser Abhandlung zusammenfassend reflektieren und die bis dahin vernachlässigten Bereiche kurz diskutieren.

Bevor jedoch der Einstieg in die Materie der Erstellung wissenschaftlicher Arbeiten erfolgen kann, noch ein Hinweis zu der in diesem Schriftstück verwendeten Anrede der Leserinnen und Leser. Zur Vermeidung eines unnötig komplizierten Satzbaus, bedingt durch eine geschlechtsspezifische Differenzierung, soll innerhalb dieser Abhandlung auf die explizite Unterscheidung in der Anrede der Leserinnen und Leser verzichtet werden. Stattdessen wird stellvertretend eine maskuline Anrede bevorzugt. Daher wird an dieser Stelle bei allen Leserinnen um Pardon für diese Entscheidung gebeten.

1.1 Status quo

Welcher Grund könnte vorliegen, warum sich jemand mit der Thematik der Erstellung wissenschaftlicher Abhandlungen auseinander setzen sollte? Diese Frage kann sicherlich von niemanden allumfassend und vor allem richtig beantwortet werden. Aber warum sollte sich dennoch jemand mit diesem Thema befassen? Weil nahezu jede lehrende Einrichtung den eigenen Vorstellungen in Bezug auf

[1] sprich *Lah-Tech*

Aufbau und Aussehen von wissenschaftlichen Abhandlungen den Vorzug einräumen, ohne dabei zu berücksichtigen, dass die Leser der publizierten Schriftsätze vor zum Teil erhebliche Herausforderungen gestellt werden!

Kurz gesagt existieren nahezu an jeder Hochschule und jeder Universität teilweise gegensätzlich lautende Leitfäden und Richtlinien zu Erstellung schriftlicher Arbeiten. Einige ausgewählte Beispiele sollen helfen, diese These zu untermauern.

Gegensätzlich lautende Richtlinien

Im Leitfaden der Christian-Albrechts-Universität zu Kiel steht zu lesen, dass Tabellen durch Überschriften zu kennzeichnen sind [29, Seite 9]. Hingegen fordert beispielsweise der Leitfaden der Fachhochschule für Oekonomie & Management, dass Tabellen mit Unterschriften zu betiteln sind [13, Seite 14].

Ein anderes Beispiel findet sich im Umgang mit so genannten Trivialabkürzungen. Beispielsweise gestattet der Leitfaden der Christian-Albrechts-Universität zu Kiel den Autoren die Verwendung gebräuchlicher Abkürzungen, wenn diese ebenfalls im Duden zu finden sind [29, Seite 9]. Indessen kann im Leitfaden der Fachhochschule für Oekonomie & Management nachgelesen werden, dass Abkürzungen „sparsam" zu verwenden sind [13, Seite 13]. Dies wiederum bedeutet, dass speziell von der Verwendung gebräuchlicher Abkürzungen abzusehen ist.

Widersprüchlich lautende Richtlinien

Ein Beispiel für in sich widersprüchliche Richtlinien findet sich an der Technischen Universität zu Chemnitz. Darin heißt es, dass ein eineinhalbfacher Zeilenabstand gewählt werden soll. Jedoch wird mit der Begründung einer „Rohstoffersparnis" direkt im anschließenden Satz empfohlen, dass der Blattrand nicht mehr als drei Zentimeter betragen sollte [23, Seite 7].

Bereits ein simpler Test belegt die Absurdität dieser Begründung. Denn nimmt man exemplarisch an, dass auf einem gängigen Blatt Papier in Größe DIN A4 bei einer gewählten Schriftgröße von zwölf Punkt und einem umgebenden Seitenrand von drei Zentimetern das kleine Wörtchen „test" bequem 1.000-mal platziert werden kann, dann bewirkt die Änderung des Zeilenabstandes auf die empfohlenen eineinhalb Zeilen eine Verlängerung des Textes um die erwarteten 50 %. Verringert man nun, bei sonst gleichen Randbedingungen, den Blattrand auf zwei Zentimeter, dann beträgt der Überstand des Testtextes immerhin noch zirka 20 %!

An der Justus-Liebig-Universität zu Gießen kann ein Leitfaden nachgeschlagen werden, der teilweise unlogisch erscheinende Forderungen enthält. Darin wird beispielsweise verlangt, dass „Tabellen und Abbildungen (...) für sich verständlich sein [müssen]!" [16, Seite 6], um diese Feststellung im nachfolgenden Satz

zu revidieren! Denn hierin erfolgt der Hinweis, dass „Tabelle“ durch „Tab.“ und „Abbildung“ durch „Abb.“ zu ersetzen sei [16, Seite 6]. Jedoch existiert innerhalb der gesamten Richtlinie kein Hinweis darauf, dass beide Abkürzungen auch in das Abkürzungsverzeichnis aufzunehmen sind. Denn es sollte nicht davon ausgegangen werden, dass einem jeden Leser die Bedeutung der Abkürzungen „Abb.“ und „Tab.“ geläufig ist.

Im weiteren Verlauf des Leitfadens der Gießener Universität steht dann auch zu lesen, dass alle Tabellen und Abbildungen über eine Quellenangabe verfügen sollen, auch wenn diese durch den Autor angefertigt wurden [16, Seite 6]. Diese Information ist redundant und somit überflüssig. Denn der Umstand, dass ein Autor eine Abbildung oder auch eine Tabelle selbst erstellt hat, spiegelt sich bereits im Fehlen einer entsprechenden Quellenangabe wider.

Nachdem die zuvor gezeigten Beispiele die Existenz absurder und widersprüchlicher Regeln offenbart haben, gilt es festzuhalten, dass diverse der vorhandenen Leitfäden und Richtlinien zur Erstellung wissenschaftlicher Abhandlungen in vielerlei Hinsicht von den Grundregeln einer guten Lesbarkeit abweichen.

Aus diesem Grund werden im weiteren Verlauf der hier vorliegenden Abhandlung die wichtigsten dieser Grundregeln erarbeitet. Der Versuch, vor allem eine sinnvolle Untermenge geltender Regeln zu bestimmen, soll ebenfalls Gegenstand der nachfolgenden Ausführungen sein.

Diese Arbeit hegt auf keinen Fall den Anspruch, Leitfäden oder Richtlinien zur Erstellung wissenschaftlicher Abhandlungen zu ersetzen. Vielmehr sollen geltende Regularien hinterfragt und im Hinblick auf ihren Beitrag zur Verbesserung der Lesbarkeit geprüft werden.

1.2 Abgrenzung

Das im Rahmen dieser Diplomarbeit implementierte *Paket zur Erstellung wissenschaftlicher Arbeiten* basiert auf dem Textsatzsystem LaTeX. Da der Hintergrund dieses Systems der breiten Masse nicht allgemein geläufig ist, erscheint eine Abgrenzung zu anderen Systemen als sinnvoll. Eine entsprechende Gegenüberstellung soll Inhalt dieses Abschnittes sein.

Führt man sich die derzeitige Marktsituation bezüglich der verfügbaren Computerprogramme zur Erstellung textorientierter Dokumente vor Augen, kann man feststellen, dass zur Erstellung der anfallenden Texte in weiten Teilen der automatisierten Industrie und in fast allen computergestützten Bereichen des Dienstleitungssektors so genannte Textverarbeitungsprogramme zum Einsatz kommen. Diese auch als *Word Processor* bezeichneten Programme sind zumeist Bestandteil eines entsprechenden Office-Pakets.

Wie bereits angedeutet, verfügen derartige Textverarbeiter über einen hohen Verbreitungsgrad, da sie allesamt dem Leitsatz *What you see is what you get* (WYSIWYG) folgen. Somit zeichnen sich Textverarbeitungen dadurch aus, dass sie von deren Anwendern in der Regel leicht und intuitiv bedient werden können. Dies begründet auch, warum sich Textverarbeitungsprogramme einer großen Beliebtheit erfreuen, obgleich das auf Papier gedruckte Ergebnis weit von den im Buchdruck gelten Qualitätsanforderungen abweicht!

Hingegen zeichnen sich Textsatzsystems, wie im hier diskutierten Fall von LaTeX, dadurch aus, dass der druckfertig ausgegebene Text nach den Erfahrungen und Regularien des Druckerhandwerks formatiert wird. Dieser Vorgang wird auch als „Setzen des Textes“ oder kurz Textsatz bezeichnet.

Also verfolgt ein Satzsystem wie LaTeX einen zu Textprogrammen völlig gegenteiligen Ansatz. Denn hierbei steht das erzeugte Ergebnis und nicht die einfache und intuitive Bedienbarkeit im Vordergrund! Das hat auch zur Folge, dass sich der gesamte Prozess der Texterstellung vollständig von der eines Textprogramms unterscheidet, wie Abbildung 1 verdeutlicht.

Abbildung 1: Erstellprozess eines LaTeX-Dokuments

Abbildung 1 zeigt auch, dass zur Erzeugung eines druckfertigen LaTeX-Dokuments eine Reihe von Werkzeugen zum Einsatz kommen. Da wäre als Erstes der Editor, ein Programm zur Bearbeitung reiner Textdateien, zu nennen, der zur

Erstellung des Rohtextes, oder besser des Manuskripts, inklusive zusätzlicher Anweisungen zur Formatierung verwendet wird.

Danach werden ein oder mehrere Programme zur Konvertierung des Rohtextes in das gewünschte Ausgabeformat angewendet, um das generierte Ergebnis anschließend auf einem Drucker auszugeben oder zum Download im Internet bereitzustellen.

Dieser durchaus als komplex zu bezeichnende Vorgang bildet im Grunde genau den Prozess ab, der während des Verlegens eines Buches stattfindet. Denn ein Autor schreibt das Manuskript, dessen Inhalt durch einen Redakteur geprüft wird. Danach überträgt ein Schriftsetzer den Inhalt des Manuskripts auf die einzelnen Buchseiten, die im Anschluss durch einen Drucker auf Papier gebracht werden.

Wie sich jeder Leser sicherlich leicht vorstellen kann, benötigt die Einarbeitung in LaTeX einen erheblichen zeitlichen Aufwand! Nicht nur, weil viele verschiedene Werkzeuge beherrscht werden müssen, sondern vielmehr aufgrund der Tatsache, dass LaTeX nahezu unendliche Möglichkeiten der Kombination der Kommandos zur Textformatierung bietet.

Dies alles wirft unweigerlich die Frage auf, warum sich überhaupt jemand die Mühe der Einarbeitung in LaTeX machen sollte? Zur Beantwortung dieser Frage reicht eine Begründung vollkommen aus. Das ausgegebene Dokument ist von einer derart hohen Qualität, dass es seinesgleichen sucht! Gerade aus diesem Grund findet man häufig auch Publikationen aus wissenschaftlichen Bereichen, die in LaTeX gesetzt wurden.

1.3 Motivation und Herausforderung

Erfahrungsgemäß ist der zeitliche Aufwand zur Einarbeitung in LaTeX im Vergleich zu gängigen Textverarbeitungen überdurchschnittlich hoch. Hinzu kommt auch, dass die Verwendung von LaTeX so manchen Student vor zum Teil beachtliche Schwierigkeiten stellt, wenn er versucht, die geltenden Richtlinien seiner Hochschule einzuhalten. Denn wie bereits erwähnt, handelt es sich bei LaTeX nicht um ein intuitiv und demzufolge leicht zu bedienendes System. Vielmehr ist das Entdecken und Erlernen neuer Kommandos zu Lösung spezifischer Formatierungsprobleme ein ständiger Bestandteil des täglichen Umgangs mit diesem System. Dennoch mehrt sich die Zahl derer, die ihren Schriftsätzen zumindest rein äußerlich eine hohe Qualität verleihen wollen.

Auch fehlt während des Studiums oft die Zeit. Dies gilt besonders für Studiengänge, die parallel zur eigentlichen Berufstätigkeit absolviert werden, zusätzlichen Aufwand zu betreiben, um sich das notwendige Wissen anzueignen, das

zur Erstellung wissenschaftlicher Abhandlungen unter Verwendung von LaTeX unumgänglich wird. Dies bezeugen verschiedene, jedoch wissenschaftlich nicht bewertbare Gespräche mit anderen Studenten. Doch war der Gehalt dieser Gespräche zumeist dadurch geprägt, dass sich die Studierenden davor scheuen, ihre Seminar- oder Abschlussarbeiten mit LaTeX zu setzen. Außerdem offenbarten diese Gespräche, dass die Verfügbarkeit einer entsprechenden „Vorlage“ das Schreiben der anstehenden Arbeiten mit LaTeX sehr erleichtern würde.

Vor diesem Hintergrund entstand der sicherlich nachvollziehbare Wunsch, die in der einjährigen Einarbeitungszeit gesammelten Erfahrungen mit anderen Kommilitonen teilen zu wollen. Dies ist gewiss Grund genug, um sich im Rahmen einer Diplomarbeit mit der Implementierung eines LaTeX-Paktes zur Erstellung wissenschaftlicher Abhandlungen auseinanderzusetzen.

Nachdem nun das Thema der Abschlussarbeit feststand, bestand die grundsätzliche Herausforderung darin, die über zwei Semester hinweg erarbeitete Basiskonfiguration in Form einer entsprechenden LaTeX-Präambel in eine für einen Endanwender bequeme Form zu überführen, welche zudem auch noch leicht zu bedienen sein sollte.

Gleichwohl zog diese Nebenbedingung einen tieferen Einstieg in das LaTeX-Kernsystem nach sich. Denn zur komfortablen Benutzung musste beispielsweise nicht nur die Frage nach einer sinnvollen Aufteilung der bereits vorhandenen Makros geklärt werden. Nein, auch die Definition zusätzlicher Kommandos, die Art der Verteilung des Implementierungsergebnisses sowie die Bereitstellung einer entsprechenden Endanwenderdokumentation waren gleichfalls ungelöst. Somit bestand in der Umsetzung dieser Punkte die eigentlichen Herauforderung der Implementierung des *Pakets zur Erstellung wissenschaftlicher Arbeiten.*

2 Dokumenterstellung

Dieser Abschnitt befasst sich mit der Beantwortung der Frage nach dem Aussehen einer „gut geschriebenen“ wissenschaftlichen Arbeit. Nachdem allgemeine und dennoch wichtige Grundsätze diskutiert wurden, gilt es, einzelne Elemente schriftlicher Ausarbeitungen näher zu beleuchteten.

2.1 Allgemeine Grundsätze

Verfasser wissenschaftlicher Arbeiten sollte immer der Unterstützung der Lesbarkeit die höchste Priorität einräumen! Denn nichts wird als unangenehmer empfunden, als einen ohnehin schwierigen Fachtext lesen zu müssen, in dem der Lesefluss ständig durch zusätzliche Hürden unterbrochen wird [21].

Daher gelten einige Grundsätze, die der Verbesserung der Lesbarkeit wissenschaftlicher Ausarbeitungen dienlich sind. Einige dieser Grundsätze, die als wesentlich und wichtig erachtet werden, sollen Gegenstand dieses Abschnittes sein.

Zeilenlänge

Die Zeilenlänge ist einer der wichtigsten Einflussfaktoren für die Lesbarkeit eines Textes. Darin sind sich alle Forscher einig. Jedoch herrscht Uneinigkeit darüber, wie viele Buchstaben pro Zeile nun sinnvoll und optimal sind. Dabei schwanken die Angaben zwischen 40 und 70 Buchstaben je Textzeile [14, 18, 21, 22, 26].

Die Augensprünge beim Lesen werden als

Sakkaden bezeichnet. Der Leser springt damit

von Fixationspunkt zu Fixationspunkt. An

diesen Stellen ist der Text am schärfsten.

Quelle: *Grundkurs Typografie und Layout* [22, Seite 102]

Abbildung 2: Beispiel für das „Lesen in Sprüngen“

Neuere Studien zeigen, dass das menschliche Auge nicht einzelne Buchstaben erfasst, sondern während des Lesevorgangs vielmehr von „Punkt zu Punkt“ springt, wie Abbildung 2 illustriert. Bei jedem Stopp werden dann die nächsten drei bis zwölf Buchstaben erfasst, um sie anschließend zu entschlüsseln und weiterzuverarbeiten [10, 19, 22].

Somit kann davon ausgegangen, dass sich der eigentliche Lesevorgang nicht buchstabenorientiert vollzieht, sondern wortorientiert abläuft. Eine Zeilenlänge von sieben bis zwölf Wörtern soll dabei die Lesbarkeit optimal unterstützen [18, 22].

Betrachtet man beide Varianten empirisch etwas genauer, kommt man zu dem Ergebnis, dass eine Zeilenlänge von acht bis zwölf Wörtern im Mittel etwa 60 bis 65 Buchstaben[2] ergeben. Daher sollte der Verfasser einer wissenschaftlichen Abhandlungen eher dem Grundsatz folgen:

Bewegt sich beim Lesen der Kopf, dann ist die Zeile zu lang!

Diese Faustregel wirkt sich nicht nur positiv auf die Ermüdung des Lesers aus, sondern verbessert auch das Auffinden des Anfangs der nächsten Textzeile erheblich [1, 9, 22, 27].

Zeilenabstand

Ein in einschlägigen Richtlinien und Hinweisen zur Erstellung wissenschaftlicher Arbeiten ebenfalls gern diskutierter Punkt ist der Abstand der einzelnen Textzeilen zueinander. Da wird schon mal ein eineinhalbfacher [13, 15] und sogar bis zu zweizeiliger [12, 21] Abstand angeraten.

Auf der anderen Seite, so hat es zumindest den Anschein, wird von einigen Autoren ein größerer Zeilenabstand bevorzugt, mit dem Effekt, dass hierdurch die Seitenzahl des Werkes künstlich erhöht wird, ohne dabei eine tatsächliche Verbesserung der Lesefreundlichkeit zu erreichen.

In beiden Fällen ist offenbar unbekannt, dass der Durchschuss, wie Schriftsetzer den Abstand der einzelnen Textzeilen zueinander bezeichnen, keine frei wählbare Größe ist, sondern vielmehr durch die Wahl der Schriftart und der Schriftgröße bestimmt wird [21].

Vom Standpunkt des Leseflusses aus betrachtet behindert ein zu geringer, aber auch ein zu groß gewählter Zeilenabstand das menschliche Auge, den Beginn der nächsten Textzeile zu finden [18, 27]. Daher wirkt sich ein unglücklich gewählter Zeilenabstand unmittelbar auf den Lesefluss und somit auf die Lesbarkeit des gesamten Textes aus. Aus diesem Grund sollten Autoren wissenschaftlicher Arbeiten eine Anpassung des durch die Schriftart bestimmten Zeilenabstandes äußerst vorsichtig handhaben.

Grundsätzlich gilt der Zusammenhang, dass eine gute Lesbarkeit durch ein Zusammenspiel der Faktoren Schriftart, Schriftgröße, Wortabstand, Zeilenlänge und Zeilenabstand erreicht wird. Dadurch bedingt kann leider keine allgemeingülte Aussage über einen optimalen Zeilenabstand getroffen werden [14].

[2] Die Anzahl der Buchstaben pro Textzeile ist von sehr vielen Faktoren abhängig und kann daher nur als ungefähre Richtgröße angesehen werden.

Schriftbild

Ein weiterer wichtiger Faktor zur Unterstützung der Lesbarkeit ist die Wahl des Schriftbildes. Hierzu zählen im engeren Sinne die verwendete Schriftart, aber auch die Schriftgröße. Im weiteren Sinne kann der Zeilenabstand ebenfalls zum Schriftbild gezählt werden. Weitere Einflussgrößen des Schriftbildes existieren, sollen jedoch im Rahmen dieser Abhandlung vernachlässigt werden [26].

Für längere Abhandlungen, zu denen wissenschaftliche Arbeiten zweifelsohne gerechnet werden können, sollte einer so genannten Serifenschrift in jedem Fall der Vorrang gewährt werden [21]. Der Grund hierfür ist in der Tat im Vorhandensein kleiner Häkchen und Füßchen zu sehen, die bei den hierzu zählenden Schriftarten existieren. Durch die Aneinanderreihung der Serifen der einzelnen Buchstaben entsteht im Gesamtbild eine „unsichtbare Linie", die das Auge während des Lesens führt. Diese Zeilenführung ist bei serifenlosen Schriftarten nicht gegeben, sodass das Auge aus der Zeile „rutschen" kann.

Mancherorts scheint es üblich zu sein, besonders große Schriften zu verwenden. Über die Beweggründe kann leider nur spekuliert werden. Fest steht jedoch, dass die Verwendung einer Schriftgröße von zum Beispiel 14 Punkt den Lesefluss negativ beeinflusst [21]. Daher wird für wissenschaftliche Arbeiten ein Schriftgrad von zehn beziehungsweise zwölf Punkt empfohlen.

Hervorhebung

Häufig wird durch die Autoren diverser Abhandlungen vom Mittel der Texthervorhebung Gebrauch gemacht. Dabei wird dieses Mittel allzu oft missbräuchlich eingesetzt, was wiederum der Unterstützung der Lesbarkeit abträglich ist [21].

Somit kommt es zuweilen vor, dass von den technisch zu Verfügung stehenden Schriftarten zehn in ein und demselben Dokument, ja sogar auf einer einzigen Seite, untergebracht werden. Dies kann durchaus als schlechter Schreibstil bezeichnet werden, da als Grundsatz die Einhaltung eines einheitlichen Schriftbildes verfolgt werden sollte.

Üblicherweise weicht die Darstellung von Eigennamen vom Grundsatz des einheitlichen Schriftbildes ab. Hierbei werden in der Regel Personennamen in Kapitälchen und Dateinamen zumeist in einer der Schreibmaschine vergleichbaren Schriftart dargestellt. Weitere Ausnahmen sind ebenfalls möglich.

Mit der Verwendung von Kapitälchen zur Hervorhebung von Personennamen soll dem Leser einer Abhandlung die besondere Ehrerbietung gezeigt werden, die ein Verfasser einer namentlich erwähnten Person entgegenbringt.

Hingegen hat die namentliche Erwähnung von Dateien in schriftlichen Ausarbeitungen einen eher technischen Touch. Aus diesem Grund werden Dateinamen auch in einer sehr technisch wirkenden Schriftart dargestellt.

Auch ist des Öfteren zu beobachten, dass Autoren ihnen besonders wichtig erscheinende Wörter durch Fettdruck oder sogar durch einfache beziehungsweise durch doppelte Unterstreichung kenntlich machen. Diese Form der Textmarkierung kann beim geneigten Leser jedoch dazu führen, sich durch den Autor bevormundet zu fühlen. Daher sollten Autoren wissenschaftlicher Arbeiten immer ein Auge darauf haben, ob eine Texthervorhebung notwendig und vor allem sinnvoll ist. Denn schließlich sollte der Leser selbst darüber befinden können, welcher Teil des Textes ihm besonders wichtig erscheint.

Linien jeglicher Art wirken sich innerhalb eines Textes besonders störend auf den Lesefluss aus, da sie die durch die Serifenschrift bewirkte Führung des Auges ablenken oder sogar unterbrechen können. Aus diesem Grund sollte gänzlich auf die Verwendung von Linien, dazu zählen neben Wortunterstreichungen auch Rahmen um Abbildungen, verzichtet werden. Von diesem Grundsatz sollte nur in ganz wenigen Ausnahmen abgewichen werden.

Zum einen ist es häufig üblich, Zeilen und Spalten von Tabellen durch Linien hervorzuheben, da durch die Tabelle selbst der Lesefluss bereits unterbrochen wurde. Dennoch sollte bei der Ausgestaltung von Tabellen darauf geachtet werden, dass weniger sehr oft mehr ist.

Zum anderen ist es ebenso häufig üblich, die Kopfzeile durch eine Linie vom restlichen Text zu separieren. Bezogen auf die Unterstützung der Lesbarkeit ist dies auch nicht weiter problematisch, da der Lesefluss ohnehin durch das Ende der vorherigen Seite unterbrochen wurde. Auch wird hierdurch der Beginn des Textteils der nächsten Seite deutlich markiert.

Die zuvor diskutierten Sachverhalte legen die Schlussfolgerung nahe, dass mit der Hervorhebung von Textpassagen äußerst sparsam umgegangen werden sollte. Wenn dennoch die besondere Markierung einzelner Wörter oder Wortgruppen als notwendig erachtet wird, dann sollte diese Kenntlichmachung stets auf eine unaufdringliche Art und Weise, zum Beispiel durch die Verwendung von schräg gestelltem Text, erfolgen.

Gliederungstiefe

Die Tiefe der einzelnen Gliederungsebenen ist ebenfalls ein nicht zu unterschätzender Punkt bei der Gestaltung wissenschaftlicher Arbeiten. Hierbei gilt, dass eine Schachtelungstiefe von mehr als drei Ebenen sich negativ auf die Lesefreundlichkeit des gesamten Werkes auswirkt.

Von diesem Grundsatz wird nur in wenigen und sehr spezifischen Abhandlungen aus dem Bereich der Rechtswissenschaften und in streng naturwissenschaftlichen Abhandlungen abgewichen, da hier nicht selten eine Herleitungskette als Beweis erforderlich ist [21].

Sollte dennoch die Unterteilung eines Abschnittes in mehrere unabhängige Unterabschnitte erforderlich sein, dann empfiehlt sich die Verwendung sinnvoller Absatzüberschriften, die über keinerlei Nummerierung verfügen und auch nicht im Inhaltsverzeichnis eingetragen werden [21]. Auf die Verwendung einer anderen Schriftart und auch einer anderen Schriftgröße sollte bei der Gestaltung dieser Unterüberschriften verzichtet werden, da „fett“ zu deren Hervorhebung völlig ausreichend ist. Der Abstand zum nachfolgenden Text sollte mit ebensoviel Bedacht gewählt werden, da andererseits der Zusammenhang zum Absatz verloren gehen könnte.

Abkürzungen

Bedingt der Inhalt eines gewählten Themas die Verwendung von Abkürzungen, dann sollte in jedem Fall darauf geachtet werden, dass die entsprechenden Erläuterungen zur Abkürzung im Text selbst vorzunehmen sind. Zudem sollten diese Erläuterungen bei der ersten Verwendung der jeweiligen Abkürzungen erfolgen. Das ist nicht nur höflich, sonder erspart dem Leser auch unnötiges Umblättern in das Abkürzungsverzeichnis und somit die Unterbrechung seines Leseflusses.

Weiterhin sollte jede wissenschaftliche Arbeit über ein Verzeichnis verfügen, in dem alle im Text vorhandenen Abkürzungen samt ihrer Erklärung aufgelistet werden. Enthält hingegen der Text keinerlei Abkürzungen, dann ist auch ein Abkürzungsverzeichnis überflüssig.

Auf die Benutzung von Trivialabkürzungen, wie beispielsweise *grds.*, *bzw.* oder auch *z. B.* sowie deren Erläuterung im Abkürzungsverzeichnis sollte in Abhandlungen mit wissenschaftlichem Anspruch immer verzichtet werden. Dies hat nicht nur etwas mit der Unterbrechung des Leseflusses zu tun, sondern wird auch dem Leser gegenüber als sehr unhöflich erachtet.

Des Weiteren besteht ein gewisses Risiko, dass die Verwendung von Trivialabkürzungen das Abkürzungverzeichis [13] und somit die Abhandlung also solches trivialisieren. Zudem verfügen die Verfasser längerer Aufsätze über genügend Zeit, alle Trivialabkürzungen auch auszuschreiben.

Fußnoten und andere stilistische Hilfsmittel

Ein immer wieder gern und heiß diskutiertes Thema ist die Verwendung von Fußnoten. Dabei liegen die Meinungen über Sinn und Zweck dieses Werkzeuges zum Teil so weit auseinander, dass kaum noch klar wird, wofür sie eigentlich gedacht sind.

Fußnoten dienen im Kern der Bereitstellung zusätzlicher Informationen, die aus verschiedenen Gründen nicht im übrigen Fließtext untergebracht werden können. Dabei gilt jedoch, dass der gesamte Text auch ohne diese Zusatzinformationen lesbar und vor allem verständlich sein sollte.

Hieraus kann geschlussfolgert werden, dass der Informationsgehalt einer Fußnote nicht notwendigerweise zum Verständnis des Gesamtzusammenhangs der entsprechenden Abhandlung beiträgt und somit getrost überlesen werden kann.

Überdies neigt der Leser beim Auftreten einer Fußnote zur abrupten Unterbrechung des Leseflusses. Daher sollte in wissenschaftlichen Abhandlungen grundsätzlich auf eine Verwendung von Fußnoten verzichtet werden.

Ein allzu oft angewendetes verbales Stilmittel ist der Verweis auf andere Kapitel oder Textpassagen. Dabei muss unterschieden werden, ob ein Hinweis auf bereits zuvor gelesene Passagen erfolgt oder ob sich der Autor auf spätere Kapitel bezieht. Grundsätzlich sind Querverweise auf bereits gelesene Abschnitte möglich und auch durchaus gängige Praxis. Jedoch wird dringend von der Verwendung der Verweise auf spätere Abschnitte abgeraten. Generell sollte man auf Querverweise dieser Art verzichten, da diese den Leser zum Umblättern animieren.

In Büchern ist häufig zu beobachteten, dass sich im Randbereich einer Seite zusätzliche Erläuterungen zum Text oder auch Begriffserklärungen befinden. Dies mag sicherlich für Buchautoren ein geeignetes Werkzeug darstellen, sollte aber vom Verfasser einer wissenschaftlichen Arbeit gemieden werden.

Verfasser längerer Abhandlungen sehen sich immer wieder mit dem Problem konfrontiert, dass Textzeilen allein und verloren entweder am Ende der vorherigen oder am Anfang der nächsten Seite auftreten. Diese als Schusterjungen und Hurenkinder bezeichneten Ausreißer trüben das gesamte Schriftbild und gelten allgemein als sehr unerwünscht. Jedoch kann diesen Erscheinungen in der Regel durch die Umformulierung des Absatzes entgegengesteuert werden.

2.2 Quellenzitate

Das Zitieren der Werke anderer Autoren ist ein ebenso wesentlicher wie häufig wiederkehrender Vorgang während des Verfassens wissenschaftlicher Abhandlungen. Aus diesem Grund werden, so hat es zumindest den Anschein, von jedweder Institution eigene Notationen zur Angabe von Zitatquellen präferiert. Die Fülle vorhandener Informationen über die verschiedenen Zitierweisen belegt diese Tatsache.

Daher wird sich dieser Abschnitt nicht mit dem „Wie“ des Zitierens auseinandersetzen, sondern sich lediglich auf den Vergleich und die Bewertung ausgewählter und gängiger Varianten der Quellenangaben im laufenden Text beschäftigen. Des Weiteren werden alle Betrachtungen vor dem Hintergrund der Lesefreundlichkeit durchgeführt. Zuvor soll jedoch einmal festgestellt werden, welche Varianten von Quellverweisen überhaupt existieren.

Zitierweisen im Überblick

Grundsätzlich kann nur zwischen einer Quellenangabe in Form eines Vollbelegs und der eines Kurzbelegs unterschieden werden. Alle weiteren Variationen können auf eine dieser beiden Methoden zurückgeführt werden, wie Abbildung 3 verdeutlicht.

Abbildung 3: Varianten gängiger Zitierweisen im Überblick

Unter der Quellenangabe mit Vollbeleg, die oft auch als traditionelle Zitierweise bezeichnet wird [11], versteht man die vollständige Angabe aller bibliographischen Daten einer zitierten Quelle im laufenden Text. Da die Fülle von Informationen nur schwerlich im Fließtext unterzubringen ist, wird hierfür üblicherweise auf das Mittel der Fußnote zurückgegriffen. Abbildung 4 zeigt beispielhaft die Verwendung einer Quellangabe in Form eines Vollbelegs.

> „Eine zur Lösung einer Aufgabe vollständige Anweisung an einen Rechner bezeichnet man als **Programm** (engl.: program); der Vorgang der Erstellung einer derartigen Anweisung heißt **Programmieren** (engl.: programming).“[1]
>
> [1] Hansen, Hans Robert; Neumann, Gustav: Wirtschaftsinformatik 1. Grundlagen und Anwendungen, 9. Auflage, Lucius & Lucius, Stuttgart et al. 2005, Seite 10.

Abbildung 4: Zitatquellenangabe mit Vollbeleg

Hingegen wird bei der Variante des Zitierens mit Kurzbeleg auf eine vollständige Angabe aller bibliographischen Daten im laufenden Text verzichtet [30]. Hierbei werden die Quellenangaben zumeist auf den Namen des Autors sowie das Erscheinungsjahr des referenzierten Werkes beschränkt, wie Abbildung 5 beispielhaft darstellt. Diese Art des Zitierens setzt jedoch voraus, dass die im Kurzbeleg angegebene Quelle zumindest einmal im vorangegangen Text mit Vollbeleg zitiert wurde. Ist dies der Fall, kann unter Umständen sogar auf ein Literaturverzeichnis verzichtet werden.

> „Eine Sammlung von gleichartigen Daten heißt **Datei** (engl.: file). Eine **Datenbank** (engl.: data base) bezeichnet eine Sammlung von mehreren Dateien, die gemeinsam verwaltet werden. Die Verwaltungsfunktionen dienen beispielsweise der Abfrage, Definition, Zugriffskontrolle und der Zugriffskoordination auf diesen Datenbestand.“[2]
>
> [2] Vgl. Hansen/Neumann (2005), S. 11.

Abbildung 5: Zitatquellenangabe mit Kurzbeleg

In natur- und ingenieurwissenschaftlichen Schriften wird häufig eine weitere Form der Quellenangabe durch Kurzbeleg verwendet [30]. Hierbei wird der Informationsgehalt des Kurzbelegs bis auf einen eindeutigen Schlüssel reduziert, der zumeist durch eckige Klammern eingeschlossen wird. Dieser Schlüssel kann dann entweder aus einer einfachen Nummer oder aus der Zusammenfassung der Abkürzung des Autorennamens und dem Erscheinungsjahr bestehen. Der Leser muss nun mit Hilfe dieses Schlüssels die zum Zitat gehörenden bibliographischen Daten im Literaturverzeichnis der Abhandlung nachschlagen, da keine Referenzierung des Werks in der Fußnote erfolgt.

Eine Abwandlung des Quellenzitats durch Kurzbeleg ist die unter dem Begriff Harvard-Notation [4] bekannte Zitierweise. Hierbei erfolgt die Quellenangabe direkt hinter dem Zitat im laufenden Text. Ein Beispiel für dieser Art der Quellenangabe zeigt Abbildung 6.

> „Wirtschaftsinformatik ist die Wissenschaft von der Entwicklung und Implementierung betriebswirtschaftlicher computergestützter Informationssysteme.“ (SCHEER 1990, S. 1).

Abbildung 6: Zitatquellenangabe in Harvard-Notation

Derartig reduzierte Quellenangaben sind jedoch im Hinblick auf eine Nachprüfbarkeit des Zitats unzureichend. Daher hat die vollständige Angabe der

bibliographischen Daten des referenzierten Werkes in Form eines am Ende der Abhandlung befindlichen Literaturverzeichnisses zu erfolgen.

Eine spezielle Form der Quellenangabe ist die unter dem Begriff der abgewandelten Harvard-Notation geläufige Zitierweise [8]. Hierbei handelt es sich um eine in der Fußnote dargestellten Variante der Harvard-Notation, die sehr häufig im Bereich der Wirtschaftsinformatik und auch in betriebswirtschaftlichen Fachbereichen angewendet wird.

> „Wirtschaftsinformatik ist die Wissenschaft von der Entwicklung und Implementierung betriebswirtschaftlicher computergestützter Informationssysteme.“[1]
>
> [1] Scheer 1990, S. 1.

Abbildung 7: Zitatquellenangabe in abgewandelter Harvard-Notation

Abbildung 7 zeigt beispielhaft die Verwendung der abgewandelten Harvard-Notation. Auch bei dieser Zitiervariante erfolgt die Angabe aller bibliographischen Informationen mit Hilfe eines am Ende der Abhandlung befindlichen Literaturverzeichnisses.

Bewertung gängiger Zitierweisen

Nachdem nun alle gängigen Varianten der Quellenangaben für Zitate kurz vorgestellt wurden, soll vor dem Hintergrund der Auswirkungen auf die Lesbarkeit eine Bewertung der einzelnen Zitierweisen erfolgen.

Eine Quellenangabe durch Vollbeleg enthält alle bibliographischen Daten des Referenzwerkes in der Fußnote. Dieser Umstand bewirkt, dass Abhandlungen, die vielfach aus wenigen Quellen zitieren, unweigerlich redundante Informationen enthalten. Diese Redundanzen bergen nicht nur die Gefahr von inhaltlichen Fehlern der jeweiligen Quellenangabe, sondern sind auch unpraktisch, da der Leser immer wieder animiert wird, den Inhalt der Fußnote zu entschlüsseln. Dies unterbricht fortwährend den Lesefluss und führt dadurch zu einer Verschlechterung des inhaltlichen Verständnisses.

Auf Grund der Problematik der angesprochenen Informationsredundanz für Quellenangaben durch Vollbeleg wird in der Praxis häufig der Kurzbeleg für weitere Nennungen eines zuvor vollständig zitierten Werkes verwendet. Da die Angabe des Kurzbelegs ebenfalls in der Fußnote erscheint, wird der Fluss des Lesens hierdurch genauso negativ beeinflusst, wie das beim Vollbeleg der Fall ist. Weiterhin wird ein Leser dazu genötigt, für detaillierte Informationen zur Quellenangabe das gesamte Schriftstück rückwärtig zu durchsuchen.

Der ständige Blick in die Fußnote und eine hieraus resultierende Leseunterbrechung spricht auch gegen die Verwendung der abgewandelten Harvard-Notation.

Ein weiterer Nachteil aller bis hierhin angesprochenen Zitierweisen besteht in der deutlichen Vermehrung der Fußnoten auf einer einzelnen Textseite. Das kann unter Umständen sogar dazu führen, dass sich auf einer Seite mehr Fußnoten finden lassen als inhaltlicher Text.

Positiv anzumerken gilt es jedoch, dass eine Nennung der Quelle in der Fußnote den Vorteil bietet, sofort und ohne umzublättern Rückschlüsse über den Hintergrund des Textes ziehen zu können, wenn den Lesern das zitierte Werk bereits bekannt ist.

Die Anwendung der Harvard-Notation löst zwar das Problem des Blicks in die Fußnote. Jedoch wird hierbei der Lesefluss durch einen verhältnismäßig hohen Informationsgehalt beeinträchtigt. Dadurch sieht sich der Leser mit der Herausforderung konfrontiert, die Fortführung des Inhalts hinter dem Zitat wieder zu finden. Denn wie bereits erwähnt, erfolgt das Lesen eines Textes anhand von Sprüngen zu frei gewählten Fixationspunkten.

Bleibt noch die überwiegend in naturwissenschaftlichen Abhandlungen vorzufindende Zitierweise zu betrachten. Aus Sicht der Verbesserung des Leseflusses ist die Reduzierung des Informationsgehalts einer Quellenangabe auf einen kurzen Schlüssel die am besten geeignete Form der Zitatmarkierung!

Der Grund hierfür liegt in der Tatsache begründet, dass durch die Kürze des Schlüssels der Lesefluss am wenigsten gestört wird. Denn hierdurch werden vom Auge des Lesers keine zusätzlichen Sprünge benötigt, um die Fortführung des Inhalts wieder zu finden. Somit ist das Überlesen einer Quellenangabe leicht möglich. Der inhaltliche Zusammenhang geht ebenfalls nicht verloren.

Dennoch sollten sich Autoren wissenschaftlicher Arbeiten darüber im Klaren sein, dass zum Beispiel die Entschlüsselung einer Quellenangabe der Form [Han05] höhere Anforderungen an das Gehirn stellt, als die Entschlüsselung einer einfachen Zahl zwischen zwei eckigen Klammern.

Zusammenfassend kann nunmehr festgestellt werden, dass alle Zitierweisen, die Informationen in der Fußnote verwenden, sich negativ auf den Lesefluss auswirken. Ebenfalls negativ in Bezug auf eine Verbesserung der Lesbarkeit wirken sich die gekürzten Angaben direkt hinter einem Zitat aus.

Somit kann geschlussfolgert werden, dass von den hier diskutierten Zitiervarianten der Kurzbeleg direkt hinter einer Zitatstelle in Verbindung mit der Verwendung eines Schlüssels mit Abstand die geringste Beeinträchtigung des Leseflusses bietet. Daher sollten Autoren wissenschaftlicher Abhandlungen der in den natur- und ingenieurwissenschaftlichen Bereichen üblichen Zitierweise den Vorzug einräumen!

Möchte ein Leser dennoch mehr über das zitierte Werk in Erfahrung bringen, dann wurde der Lesefluss bereits unterbrochen und einem Blick in das Literaturverzeichnis steht nichts mehr im Wege.

2.3 Objektbeschriftungen

Wissenschaftliche Abhandlungen enthalten sehr häufig Objekte, die dem Leser spezielles Wissen in kompakter Form darbieten. Diese Darstellungen werden unter dem Begriff „Gleitobjekt“ zusammengefasst. Hierzu zählen unter anderem jegliche Form von Abbildungen, aber auch alle Arten von Tabellen. Codelistings können ebenfalls als eine Form von Gleitobjekten betrachtet werden.

Wie der Name bereits vermuten lässt, ist die Platzierung eines Gleitobjekts innerhalb des gedruckten Textes nicht exakt vorhersagbar. Daher sollten Autoren wissenschaftlicher Arbeiten immer auf eine direkte Bezugnahme, wie zum Beispiel „wie folgende Abbildung zeigt“ oder auch „stellt vorherige Abbildung dar“, verzichten. Stattdessen sollte eine indirekte Referenzierung über die Nummer des jeweiligen Gleitobjektes bevorzugt werden.

Da Gleitobjekte nicht unmittelbar zum Fließtext einer Abhandlung gehören, widerfährt ihnen auch eine Sonderbehandlung. Denn alle Gleitobjekte verfügen, mit nur ganz wenigen und spezifischen Ausnahmen, über eine Beschriftung. Dabei ist die Position der Beschriftung vielen Autoren wissenschaftlicher Abhandlungen nicht immer ganz geläufig.

Dieser Zusammenhang soll nun im Folgenden hergeleitet werden. Zu diesem Zweck wird sich jedoch auf Abbildungen, Tabellen und Codelistings, also die gängigen Gleitobjektformen einer wissenschaftlichen Arbeit in Bereichen der Informationstechnologie, beschränkt.

Beschriftung von Abbildungen

Abbildungen enthalten zumeist komplexe Zusammenhänge in graphisch aufbereiteter Form. Diese grundlegende Eigenschaft hat zur Folge, dass Abbildungen in den meisten Fällen auf einer einzelnen Dokumentseite untergebracht werden können. Dieser Umstand bewirkt, dass Abbildungen in der Praxis eine unterhalb befindliche Beschriftung erhalten.

Weiterhin stammen Abbildungen oft aus urheberrechtlich geschützten Quellen, sodass in diesem Fall einer Abbildung noch eine Quellenangabe hinzugefügt werden muss. Da dieser Teil der Beschriftung der eigentlichen Abbildung zuzurechnen ist, befindet sich der Verweis auf den Urheber auch direkt unterhalb der jeweiligen Abbildung, also genau zwischen Abbildung und Abbildungsbeschriftung.

Beschriftung von Tabellen und Codelistings

Tabellen, aber auch Quellcodelistings, haben die Eigenschaft, dass sie in der Regel über deutlich mehr textuelle Informationen verfügen, als das bei Abbildungen der Fall ist. Daher ist es nicht selten, dass dieser Typus Gleitobjekt sich über mehr als eine Seitenlänge erstreckt.

Nun könnte angenommen werden, dass Tabellen und auch Codelistings über eine Beschriftung unterhalb verfügen, wie dies bei Abbildungen der Fall ist. Dem ist aber nicht so! Denn grundsätzlich erhalten alle Gleitobjekte, mit Ausnahme der Abbildungen, ihre Beschriftung oberhalb[3].

Das wirft unweigerlich die Frage nach dem „Warum“ auf. Der Grund hierfür ist ebenso leicht verständlich wie plausibel. Der Lesefluss wird durch die Tatsache der bloßen Existenz eines Gleitobjektes unterbrochen. Somit will der Leser unmittelbar wissen, wodurch sein Fluss unterbrochen wurde. Hierfür ist die Beschriftung oberhalb eines Gleitobjekts sehr hilfreich.

Dazu kommt auch, dass innerhalb des Textes auf Tabellen und alle anderen Gleitobjekte Bezug genommen wird. Um nun dem Leser ein leichtes Auffinden des jeweiligen Objektes zu ermöglichen, ist hierfür eine einleitende Beschriftung sehr hilfreich. Er muss also nicht erst umständlich das Objektende suchen. Somit wird das Gesamtverständnis des Fachtextes durch eine Objektüberschrift positiv beeinflusst.

Ein ebenfalls wichtiger Aspekt ist die Angabe des Ursprungs des Gleitobjektes. Dabei hat die Angabe der Quelle grundsätzlich unterhalb des Objekts zu erfolgen. Dies gilt im Speziellen für Tabellen, aber auch im Allgemeinen für alle anderen Gleitobjekte, die zur Untermauerung der eigenen Thesen aus anderen Arbeiten entnommen wurden.

Codelistings weichen in wissenschaftlichen Arbeiten oft von diesem Grundsatz ab, da sie zumeist aus der eigenen Feder stammen und demzufolge die Erstellung des Codes, zumindest teilweise, Inhalt der entsprechenden Abhandlung ist. Aber auch hier gilt: wurde der Code einer Fremdquelle entnommen, dann hat die Angabe des Ursprungs unterhalb des Listings zu erfolgen.

Regel für Objektbeschriftung

Zusammenfassend kann festgestellt werden, dass eine Beschriftung dem Gleitobjekt vorangestellt wird. Dies begründet sich durch die Tatsache, dass hierdurch zum einen dem Leser direkt mitgeteilt wird, warum sein Lesefluss unterbrochen wurde und zum anderen wird hierdurch ein schnelleres Auffinden des jeweiligen Objektes ermöglicht.

[3] Die Feststellung, dass Tabellen über eine Überschrift verfügen, steht im Widerspruch zu den an der Fachhochschule für Oekonomie und Management (FOM) anerkannten, allgemeinen Richtlinien zur Erstellung wissenschaftlicher Arbeiten [13, Seite 14].

Einzige Ausnahme von diesem Grundsatz bilden Abbildungen, da diese auf einer Textseite Platz finden und somit deren Beschriftung auf einen Blick wahrgenommen werden kann.

Die Quellenangabe hat in jedem Fall unterhalb des entsprechenden Objektes zu erfolgen, da sie als eine Art Unterschrift angesehen werden kann!

2.4 Dokumentstruktur

Ein ebenfalls wesentlicher Gesichtspunkt bei der Erstellung wissenschaftlicher Abhandlungen ist deren Struktur. Jedoch sollte diese Struktur nicht willkürlich gewählt werden, sondern vielmehr allgemeinen und anerkannten Grundsätzen folgen, die überdies noch außerordentlich hilfreich für den Leser sind.

Im Allgemeinen lassen sich schriftliche Ausarbeitungen jeglicher Art in drei voneinander getrennte Bereiche unterteilen. Hierbei handelt es sich im Einzelnen um

- den Beginn des Dokuments,
- den Textteil und um
- das Ende des Dokuments.

Dabei sind Dokumentbeginn und Dokumentende vom Vorhandensein diverser Verzeichnisse geprägt. Die Diskussion über den Sinn und Zweck dieser Verzeichnisse soll Gegenstand der hieran anschließenden kurzen Einführung in das Thema der Dokumentstruktur sein.

Zweck eines Verzeichnisses

Wozu nützt dem Leser ein Verzeichnis? Verzeichnisse dienen in erster Linie der kompakten Darstellung spezieller Inhalte. Daher eignen sie sich sehr gut als zusätzliche Informationsquelle und bieten somit dem Leser einen nicht unerheblichen Mehrwert. Denn der Leser kann durch den Blick in ein Verzeichnis direkt zur Seite wechseln, auf dem sich das Objekt seines Interesses finden lässt.

Das setzt jedoch voraus, dass in den Verzeichnissen entsprechende Verweise auf die einzelnen Seiten vorhanden sind. Folgerichtig kann festgehalten werden, dass Verzeichnisse in der Regel über die Angabe einer Seitenzahl verfügen. Ansonsten bestünde für den Leser auch kein signifikanter Vorteil durch ein Verzeichnis.

Somit besteht die Aufgabe eines Verzeichnisses darin, dem Leser einer wissenschaftlichen Ausarbeitung das schnelle Auffinden einer bestimmten Information oder Sachverhaltes zu erleichtern beziehungsweise erst zu ermöglichen.

Platzierung eines Verzeichnisses

Die nächste Frage, die im Rahmen dieses Abschnittes betrachtet werden soll, ist die Frage, warum sich Verzeichnisse am Beginn oder am Ende einer Abhandlung befinden. Zur Beantwortung dieser Frage soll nun das Verhalten eines Leser etwas genauer betrachtet werden.

Ein Leser trifft auf ein unbekanntes Buch

Man stelle sich vor, man erhält ein unbekanntes Buch. Nachdem nun der Titel weiteres Interesse geweckt hat, möchte man mehr über den Inhalt in Erfahrung bringen. Dazu wird das betreffende Buch aufgeschlagen und man findet als Erstes das Inhaltsverzeichnis!

Ein Inhaltsverzeichnis ermöglicht dem Leser, neben der inhaltlichen Übersicht, auch etwas über die Struktur und die Aufteilung des Werkes in Erfahrung zu bringen. Dieser Umstand ist für den Leser mehr als hilfreich, da er auf einen Blick entscheiden kann, eventuell tiefer in das jeweilige Thema einzusteigen.

Dieser Zusammenhang belegt die in der Praxis vorzufindende Tatsache, dass Verzeichnisse, welche die Struktur einer Abhandlung widerspiegeln, sich auch am Anfang dieser Abhandlung befinden.

Das Inhaltsverzeichnis bildet zweifelsfrei die Struktur einer Ausarbeitung ab. Jedoch ist dies bei einem Abkürzungs-, einem Abbildungs- und auch einem Tabellenverzeichnis, die sich ebenfalls am Beginn eines Dokuments befinden sollen, nicht so eindeutig gegeben.

Vielmehr erweckt die Praxis den Anschein, dass diese Verzeichnisse nach „Gutdünken“ in den jeweiligen Abhandlungen verteilt werden. Da findet man schon mal Bücher, in denen die Abbildungen und Tabellen im Anhang gelistet werden. Ist das sinnvoll? Oder ist es doch besser, Abkürzungen, Abbildungen und Tabellen an den Anfang des literarischen Werkes zu stellen?

Diese Frage kann leider nicht hinreichend genau beantwortet werden! Somit muss sich ein Autor oft, zumindest in diesem Punkt, auf seinen eigenen guten Geschmack stützen, um die Entscheidung für eine Platzierung dieser Verzeichnisse treffen zu können.

Aber dies betrifft vorwiegend Buchautoren. Denn die Autoren wissenschaftlicher Arbeiten unterliegen nicht selten den Regularien ihres Fachgebiets. Und hier herrscht die einhellige Meinung, dass Abbildungen, Tabellen und auch Abkürzungen am Beginn der Ausarbeitung zu finden sind [4, 13, 15, 5].

Ein Leser trifft auf ein bekanntes Buch

Betrachtet man nun den Umstand, dass ein Leser zu einem ihm bekannten Buch greift, beispielsweise um etwas nachzuschlagen, dann kann es für ihn durchaus

hilfreich sein, darin Verzeichnisse mit nützlichen Informationen vorzufinden. Als Vertreter eines solchen Verzeichnisses sei der Index hier namentlich erwähnt.

Denn unter der Voraussetzung, dass der geneigte Leser den Inhalt einer Abhandlung bereits kennt, ist das schnelle Auffinden einer dort enthaltenen, spezifischen Information außerordentlich zielführend.

Da derartige Verzeichnisse häufig sehr viele dieser nützlichen Informationen enthalten und demzufolge recht lang sein können, empfiehlt es sich, diese auch am Ende des Werkes unterzubringen.

Manch einer wird sich sicherlich fragen, warum diese Verzeichnisse nicht doch besser am Anfang platziert werden sollten. Die Antwort darauf ist recht einfach und einleuchtend zugleich. Ein Index als Beispiel enthält überhaupt keine Informationen über die Struktur der betreffenden Abhandlung. Er bietet somit dem Leser auch keinen zusätzlichen Nutzen, wenn er das Buch zum ersten Mal aufschlägt.

Zudem kommt auch, dass sich zum Beispiel ein Index nicht selten über viele Seiten erstreckt. Stünde der Index am Anfang, dann würde dies die Suche nach dem inhaltlichen Beginn der Abhandlung noch zusätzlich erschweren.

Alle zuvor diskutierten Zusammenhänge legen die Schlussfolgerung nahe, dass sich am Anfang einer wissenschaftlichen Ausarbeitung nur Verzeichnisse befinden, die den strukturellen Aufbau der Arbeit widerspiegeln. Hingegen befinden sich am Ende der Arbeit nur solche Verzeichnisse, die ein schnelles Auffinden von Informationen innerhalb des Dokuments ermöglichen.

2.4.1 Dokumentbeginn

Der einleitende Teil einer wissenschaftlichen Arbeit besteht in der Regel aus einem Vorspann und einer strukturellen Übersicht in Form diverser Verzeichnisse. Die genauere Betrachtung der einzelnen Komponenten des Dokumentenbeginns soll Gegenstand dieses Abschnittes sein.

Titelseite

Die Titelseite ist die erste Seite einer wissenschaftlichen Arbeit! Diese Trivialaussage zielt darauf ab, den geneigten Leser dafür zu sensibilisieren, dass der Vorspann einer wissenschaftlichen Arbeit nicht nur aus einem einzelnen Blatt mit Angaben zum Institut, Thema und Autor besteht, sondern auch weitere Textpassagen bereithalten kann.

Da wäre zunächst einmal der Sperrvermerk zu nennen. Er wird vor allem bei Diplomarbeiten verwendet, um festzulegen, dass der Inhalt der Abhandlung ausdrücklich nur einem befugten Personenkreis zugänglich sein darf. Also ist

ein Sperrvermerk häufig in den wissenschaftlichen Arbeiten zu finden, die auf Basis firmenspezifischer und zumeist geheimer Daten beruhen [13]. Denn die Preisgabe derartiger Informationen könnte unter Umständen die Existenz des jeweiligen Unternehmens gefährden.

Sollte nun aufgrund des behandelten Themas ein Sperrvermerk für wissenschaftliche Abhandlungen erforderlich werden, dann empfiehlt es sich, diesen unbedingt auf der Seite direkt hinter dem Titelblatt zu platzieren!

Ein Vorwort und auch eine Danksagung sind ebenfalls potenzielle Kandidaten für den Vorspann einer wissenschaftlichen Abhandlung. Sie werden jedoch der Arbeit nur bei Bedarf beigefügt.

All diese zusätzlichen Seiten weisen die Gemeinsamkeit auf, dass sie typischerweise vor dem Inhaltsverzeichnis zu platzieren sind und somit von einer Seitennummerierung ausgenommen werden. Dies ist durch die Tatsache begründet, dass derartige Seiten nicht zum eigentlichen Inhalt der Abhandlung gezählt werden können!

Inhaltsverzeichnis

Das Inhaltsverzeichnis ist das erste Verzeichnis, das sich an den Vorspann einer wissenschaftlichen Arbeit anschließt. Es ist für die Abhandlung von immanenter Wichtigkeit, da es neben der inhaltlichen Übersicht auch die Gesamtstruktur des Werkes widerspiegelt.

Weitere Verzeichnisse

Sollte die wissenschaftliche Abhandlung, wie in den meisten Fällen üblich, auch Tabellen, Abbildungen oder Abkürzungen enthalten, dann sind durch den Autor entsprechende Verzeichnisse bereitzustellen. Dabei sind diese Verzeichnisse direkt hinter dem Inhaltsverzeichnis zu positionieren.

Leider kann die Reihenfolge dieser Verzeichnisse nicht eindeutig festgelegt werden, da sie in den jeweils geltenden Regelungen zur Erstellung wissenschaftlicher Arbeiten variieren [13, 15, 5]. Dennoch erscheint eine Reihenfolge von

- Abkürzungsverzeichnis,
- Abbildungsverzeichnis und
- Tabellenverzeichnis

als sinnvolle Wahl, da diese der Häufigkeit ihrer Benutzung am nächsten kommt. Vermutlich werden die Leser einer wissenschaftlichen Abhandlung das Abkürzungsverzeichnis viel öfter frequentieren als beispielsweise das Abbildungs- oder

das Tabellenverzeichnis. Daher bietet es sich förmlich an, das Abkürzungsverzeichnis direkt hinter dem Inhaltsverzeichnis zu platzieren.

Seitenvorschub bei Verzeichnissen

Grundsätzlich sollte sich jedes einzelne Verzeichnis auf einer separaten Seite befinden. Jedoch führt dies im Falle sehr kurzer Verzeichnisse zu einem weniger schönen Aussehen des Dokumentbeginns. In diesem Fall kann ein deutlich besseres Layout der Arbeit durch die Zusammenfassung der einzelnen, kurzen Verzeichnisse auf fortlaufenden Seiten erreicht werden.

Diese Ausnahmeregel gilt jedoch nicht für das Inhaltsverzeichnis! Denn diesem Verzeichnis wird eine gewisse Sonderrolle zuteil und es beginnt somit immer auf einer neuen Textseite. Alles Weitere muss dann ebenfalls auf einer neuen Seite beginnen.

2.4.2 Textteil

Wie sich leicht vorstellen lässt, ist das Treffen allgemeingültiger Aussagen über den inhaltlichen Abschnitt einer wissenschaftlichen Arbeit ein eher schwieriges Unterfangen. Daher kann und muss sich der Inhalt dieses Kapitels auf einige wenige grundsätzliche Fragen beschränken.

Textzwischenraum

Wie bereits angesprochen, ist eine isolierte Betrachtung des idealen Zeilenabstandes nur sehr schwer möglich, da eine gute Lesbarkeit von gedrucktem Text durch das Zusammenspiel der Faktoren Schriftart, Schriftgröße, Wortabstand, Zeilenlänge und Zeilenabstand bestimmt wird [22, 26].

Grundsätzlich kann davon ausgegangen werden, dass die meisten verfügbaren Schriftarten die Anforderungen für eine gute Lesbarkeit bezüglich ihrer horizontalen und vertikalen Abstände bereits erfüllen. Daher ist in der Regel auch ein einzeiliger Abstand für wissenschaftlichen Abhandlungen ausreichend. Sollte der Text durch die verwendete Schrift dennoch gedrungen wirken, dann ist zu dessen Auflockerung ein etwa 1,2-zeiliger Abstand empfehlenswert.

An dieser Stelle soll jedoch nicht verheimlicht werden, dass die zuvor gegebene Empfehlung im Grunde nur ihre Gültigkeit für auf Papier gedruckte Texte besitzt. Das liegt daran, dass ein zu wählender Zeilenabstand nicht ausschließlich von der Schrift abhängt, sondern auch durch das eingesetzte Anzeigemedium tangiert wird. Beispielsweise existieren Empfehlungen, dass Texte, die vorwiegend am Bildschirm gelesen werden, eine Verbesserung der Lesbarkeit durch einen Abstand von mindestens eineinhalb Zeilen erfahren [18, 27].

Dieser kurze Exkurs deutet eine inhärente Komplexität an, auf deren tief greifende Diskussion bezüglich der Lesbarkeit von Bildschirmtexten innerhalb dieser Abhandlung leider verzichtet werden muss. Für den interessierten Leser, der mehr über eine Beeinflussung des Leseflusses durch elektronische Medien in Erfahrung bringen möchten, sei auf die Existenz verschiedentlicher Studien zu diesem Thema hingewiesen.

Seitenvorschub nach Kapitelende

Jeder, der bereits einmal ein Buch in der Hand hielt, konnte feststellen, dass die Kapitel der Hauptebenen auf einer neuen Seite beginnen. Häufig ist auch zu beobachten, dass Hauptkapitel sogar nur auf den ungeraden Seiten ihren Anfang nehmen. Das hat einen guten Grund! Denn wurde ein Kapitel beendet, dann wurde auch ein gewisser Themenkomplex abgeschlossen. Der Leser hat nun die Gelegenheit, den Inhalt des Kapitels zu reflektieren. Der Lesefluss ist somit unterbrochen.

Also stellt sich die Frage, warum nicht auch in wissenschaftlichen Abhandlungen jedes Hauptkapitel auf einer neuen Seite beginnen sollte. Die Antwort ist simpel. Auch in wissenschaftlichen Arbeiten sollten die Kapitel der Hauptgliederungsebenen auf einer neuen Seite beginnen. Das sieht nicht nur besser aus, sondern erleichtert auch dem Leser die Suche nach einem bestimmten Kapitel.

Nebenbei sein noch angemerkt, dass der Seitenvorschub am Kapitelende den verschiedenen Textverarbeitungswerkzeugen die Formatierung der Kopfzeile stark erleichtert, da hierdurch Konflikte zwischen konkurrierenden Kapitelüberschriften vermieden werden können.

Allgemeiner Aufbau

Wie bereits einleitend erwähnt, kann über den Inhalt einer wissenschaftlichen Ausarbeitung keine allgemeingültige Aussage getroffen werden, da dieser ausschließlich vom bearbeiteten Thema abhängt. Dies ist für den grundlegenden Aufbau der Arbeit dennoch möglich. Denn im Kern kann fast jede Abhandlung mit wissenschaftlichem Hintergrund bis auf vier Bereiche abstrahiert werden. Bei diesen Bereichen handelt es im Einzelnen um

- die Einleitung,
- den Grundlagenteil,
- den Kern der Abhandlung und um
- die Schlussbetrachtung.

Die Einleitung der Abhandlung

Der Teil der Einleitung ist einer der wichtigsten Bereiche der gesamten Abhandlung, da hier sämtliche Randbedingungen vereinbart und festgelegt werden können. Dazu zählen unter anderem die Definition der Zielgruppe, die Abgrenzung des Themas und die Motivation des Autors.

Weiterhin besteht die Möglichkeit, eine Auswertung und auch eine Bewertung der verwendeten Literatur sowie deren Quellen vorzunehmen. Dies ist zum Beispiel dann sehr sinnvoll, wenn der Autor über ein bisher stark vernachlässigtes oder auch über ein brandneues Thema schreibt und demzufolge die verfügbaren Literaturquellen nur sehr spartanisch ausfallen.

Die Unterbringung historischer Hintergrundinformationen, wenn sie zur allgemeinen Aufhellung des Kernthemas hilfreich sind, können ebenfalls in der Einleitung der wissenschaftlichen Arbeit untergebracht werden.

Die Grundlagen einer Abhandlung

Im Teil der Grundlagen werden in der Regel vom Autor die Basisinformationen bereitgestellt, die zum Verständnis des im Abhandlungskern befindlichen Teils der Arbeit notwendig sind. Dabei sollte sich das vermittelte Basiswissen an den Bedürfnissen der Zielgruppe orientieren. Es ist sicherlich überflüssig, einer aus Informatikern bestehenden Zielgruppe den grundsätzlichen Aufbau eines Computers zu erläutern.

Der Kern einer Abhandlung

Der inhaltliche Abschnitt der wissenschaftlichen Arbeit besteht zumeist aus der Beschreibung einer zur Abhandlung gehörenden „Durchführung". Dieser stark themenspezifische Teil kann beispielsweise die Implementierung einer Problemlösung oder auch die Analyse sowie die Auswertung einer Umfrage enthalten.

Je nach Art der Abhandlung kann dessen Kern aus einem oder mehreren Kapiteln bestehen. Dabei hängt die Anzahl der Kapitel zumeist von der bearbeiteten Aufgabe ab. Jedoch sollte in einer wissenschaftlichen Arbeit für diesen Komplex mindestens ein Kapitel spendiert werden.

Die Schlussbetrachtung

Eine Schlussbetrachtung, oder auch ein Fazit, sollte in einer wissenschaftlichen Abhandlung niemals fehlen, da sie die Reflexion und Bewertung der eigenen Ergebnisse zulässt. Sie bietet somit auch Raum zur Selbstkritik an diesen Ergebnissen.

Des Weiteren stellt die Schlussbetrachtung eine Plattform zur Benennung ungelöster Probleme und unbeantworteter Fragen dar. Ein Ausblick in zukünftige Entwicklungen wird ebenfalls gern in diesem Abschnitt untergebracht.

2.4.3 Dokumentende

Der Abschluss einer wissenschaftlichen Arbeit ist ebenfalls durch die Existenz verschiedentlicher Verzeichnisse gekennzeichnet. Jedoch finden sich hier zumeist nur Verzeichnisse wieder, die für den Leser nützliche und hilfreiche Zusatzinformationen bereithalten. Die detaillierte Betrachtung der am Ende einer wissenschaftlichen Abhandlung vorzufindenden Verzeichnisse soll Gegenstand dieses Abschnittes sein.

Abgrenzung

Oft werden die Verzeichnisse des Dokumentenendes lapidar als Anhang bezeichnet. Doch trifft dies bei genauerer Betrachtung nur bedingt zu. Denn der Anhang im eigentlichen Sinne ist der Teil einer Publikation, der inhaltlich zur Abhandlung selbst gehört [2, 12, 7].

Hingegen hat zum Beispiel ein Literaturverzeichnis die Aufgabe, den Nachweis zu den durch den Autor aufgestellten Behauptungen zu erbringen und somit eine Nachprüfbarkeit der Arbeit zu ermöglichen. Also gehört ein Literaturverzeichnis nicht unmittelbar zum Abhandlungsinhalt.

Dieser Vergleich ermöglicht nun die grundsätzliche Unterscheidung, welches der in wissenschaftlichen Arbeiten gängigen Verzeichnisse in einen Anhang gehören und welche davon separiert werden.

Anhänge

Wie zuvor bereits festgestellt, dient der Anhang dazu, zusätzliche, zum Inhalt der Arbeit gehörende Elemente aufzunehmen, weil sie beispielsweise für eine Unterbringung im Kerntext zu umfangreich sind. Typische Vertreter derartiger Elemente sind Statistiken sowie lange, oft mehrseitige Tabellen, ganzseitige Abbildungen, Musterformulare, Fragebögen und auch Codelistings.

A Basis und Ergebnisse der Umfrage

A.1 Verwendeter Fragebogen

...

A.2 Erhobene statistische Daten

...

A.3 Diagramme der Auswertung

Abbildung 8: Beispielanhang einer Abhandlung

Die Sonderstellung des Anhangs innerhalb einer wissenschaftlichen Arbeit wird dadurch betont, dass die Abschnitte ebenfalls in die Kapitelnummerierung ein-

bezogen werden. Nur wird dabei in der Regel die Nummerierung der einzelnen Ebenen durch einen Buchstaben eingeleitet [6, 5]. Somit könnte sich beispielsweise ein Anhang einer Abschlussarbeit, basierend auf der Auswertung statistischer Erhebungen, wie in Abbildung 8 dargestellt, ergeben.

Bei sehr umfangreichen Anhangsinhalten kann es unter Umständen sinnvoll sein, einem Anhang ein eigenes Verzeichnis voranzustellen [6]. Dies erleichtert dem Leser die Suche nach speziellen Inhalten des Anhangs.

Verzeichnisse

Die Verzeichnisse des Dokumentenendes gehören nicht unmittelbar zum Inhalt der wissenschaftlichen Abhandlung, sondern dienen dem Leser vielmehr als zusätzliche Informationsquelle.

Häufige Vertreter dieser Verzeichnisse sind das Literaturverzeichnis, das Glossar und auch der Index. Alle diese Verzeichnisse sind grundsätzlich als optional zu betrachten. Doch wenn sie in einer Abhandlung verwendet werden, dann weisen sie alle die Gemeinsamkeit auf, dass sie über keinerlei Nummerierung ihrer Kapitelüberschriften verfügen.

Literaturverzeichnis

Unter der Voraussetzung, dass entweder keine Fremdliteratur verwendet wurde oder alle Literaturverweise durch Vollbeleg bereits im laufenden Text abgedeckt wurden, kann zum Beispiel auf ein Literaturverzeichnis verzichtet werden. Jedoch ist diese Art der Betrachtung von eher akademischer Natur, da sie für wissenschaftliche Arbeiten in der Regel ausgeschlossen werden kann. Somit ist das Literaturverzeichnis als Pflichtbestandteil einer Abhandlung mit wissenschaftlichem Hintergrund zu sehen.

Glossar und Index

Anders verhält es sich hingegen bei einem Glossar, das auch als Sachwortverzeichnis geläufig ist, und einem Index, der häufig auch als Stichwortverzeichnis bezeichnet wird. Denn die Erstellung beider Verzeichnisse bedingt zusätzliche Anstrengungen des Autors, die zum Teil erheblichen Aufwand bedeuten können. Diese Betrachtung ist jedoch im Zeitalter der elektronischen Datenverarbeitung von eher untergeordneter Bedeutung, da eine Verzeichniserstellung in der Regel automatisiert erfolgen kann.

Somit kann geschlussfolgert werden, dass eine „gut geschriebene“ wissenschaftliche Abhandlung zumindest über einen Index zum Nachschlagen bestimmter und wichtiger Begriffe verfügen sollte. Auf die Bereitstellung eines Glossars zur Erläuterungen spezieller Fachbegriffe sollte in wissenschaftlichen Arbeiten ebenfalls nicht verzichtet werden.

Eidesstattliche Versicherung

Ein gerade für Diplomarbeiten überaus wichtiger Gesichtspunkt ist das einem Schwur gleichgestellte Versprechen des Autors, alle Inhalte seiner Arbeit selbst verfasst und alle Zitate entsprechend gekennzeichnet zu haben. Diese, auch als „Ehrenwörtliche Erklärung“ bezeichnete, Eidesstattliche Versicherung kommt der Unterschrift des Autors gleich und hat sich somit auf jeden Fall am Ende der wissenschaftlichen Abhandlung zu befinden.

Festlegung der Reihenfolge

Eine genau definierte Reihenfolge der Verzeichnisse am Ende einer wissenschaftlichen Arbeit konnte nicht nachvollzogen werden, da sich in diesem Punkt die Meinungen an den jeweiligen Präferenzen zu orientieren scheinen [2, 6, 13, 15]. Zur Klärung dieser oft umstrittenen Frage kann jedoch eine ganz pragmatische Betrachtungsweise herangezogen werden.

Betrachtet man den Inhalt eines speziellen Verzeichnisses etwas genauer, dann kann festgestellt werden, dass dessen Inhalt einen stärkeren oder schwächeren Bezug zum Inhalt der restlichen Arbeit aufweist. Somit kann die Stärke dieses Bezugs als Indikator zur Festlegung der Position des betreffenden Verzeichnisses herangezogen werden.

Anhang — Wie zuvor oben angedeutet, beinhaltet der Anhang nur die Informationen, die aus verschiedenen Gründen im Kerntext nicht untergebracht werden konnten. Hierdurch besteht eine sehr stark ausgeprägte Bindung zum Inhalt der restlichen Abhandlung und daher sollte sich der Anhang direkt an den Textteil anschließen.

Literatur — Da ein Literaturverzeichnis zum Nachweis und zur Nachvollziehbarkeit der aufgestellten Thesen dient, ist hierdurch eine stark ausgeprägte Bindung zum restlichen Inhalt festzustellen. Aus diesem Grund sollte sich das Literaturverzeichnis an den Anhang anschließen.

Glossar — Das Glossar erläutert spezielle Begriffe, auf die sich innerhalb der Abhandlung bezogen wurde. Diese Begriffserklärungen sind aber zum Verständnis der eigentlichen Abhandlung nicht zwingend erforderlich, zumindest sollte dies so sein. Daher ist für ein Glossar eine auch eher schwache Bindung zum restlichen Inhalt festzustellen und somit sollte das Glossar dem Literaturverzeichnis folgen.

Index Ein Index listet bestimmte Begriffe einer Abhandlung in alphabetischer Reihenfolge, ohne dabei auf bestimmte Bedeutungen Rücksicht zu nehmen. Daher kann davon ausgegangen werden, dass die Bindung des Indexes zum restlichen Inhalt am schwächsten von allen anderen ausfällt und daher sollte der Index dem Glossar folgen.

Versicherung Für die Eidesstattliche Versicherung könnte die zuvor gezeigte Argumentationskette ebenfalls fortgeführt werden, jedoch ist ihre Bedeutung eine andere. Denn die Eidesstattliche Versicherung ist die Erklärung des Autors, dass er nicht „abgeschrieben“ hat und wirkt damit wie eine Unterschrift. Und eine Unterschrift steht nun mal unter einem Dokument. Somit muss die ehrenwörtliche Erklärung der letzte Bestandteil einer wissenschaftlichen, respektive einer Diplomarbeit sein!

Seitenvorschub bei Verzeichnissen

Grundsätzlich gilt auch für die Verzeichnisse des Dokumentenendes das bereits in Kapitel 2.4.1 auf Seite 23 Gesagte. Jedoch sind die hier befindlichen Verzeichnisse häufig sehr ausladend. Daher wird in der Praxis zumeist darauf verzichtet, die am Ende einer wissenschaftlichen Abhandlung vorhandenen Verzeichnisse auf fortlaufenden Seiten unterzubringen.

Dies gilt aber auf keinen Fall für die Eidesstattliche Versicherung. Denn diese hat sich immer auf der letzten Seite der Abhandlung zu befinden und sie ist stets von allen anderen Seite separiert zu betrachten!

2.5 Seitennummerierung

Dieser Abschnitt greift erneut die einzelnen Komponeten einer wissenschaftlichen Arbeit auf und betrachtet dabei die Nummerierung der einzelen Seiten. Diese Betrachtungen können und sollten losgelöst von allen anderen Blickwinkeln geschehen, da eine Seitennummerierung bis auf wenige wichtige Zusammenhänge abstrahiert werden kann.

Dokumentbeginn

Wie bereits geschildert, besteht der Beginn eines wissenschaftlichen Dokuments aus einem Vorspann, der von verschiedentlichen Verzeichnissen mit römischer Seitennummerierung, beginnend bei eins, gefolgt wird [12, 13, 5]. Tabelle 1 zeigt eine Übersicht der einzelnen Bestandteile in Abhängigkeit von ihrer jeweils anzuwendenden Seitennummerierung.

Textteil und Dokumentende

Beginnend mit dem ersten Abhandlungskapitel, also zu Anfang des Textteils, wird von der zuvor römischen auf eine arabische Seitennummerierung gewechselt [12, 13, 5], die ebenfalls bei eins beginnt.

Die zu diesem Zeitpunkt vorgenommene Änderung der Art der Nummerierung verfolgt den Zweck, dem Leser deutlich mitzuteilen, dass an dieser Stelle nun mit dem inhaltlichen Teil begonnen wird.

Tabelle 1: Seitennummerierung des Dokumentbeginns

Komponente	Bestandteile	Seitennummerierung
Vorspann	Titelseite	**keine**
	Sperrvermerk	**keine**
	Danksagung	**keine**
	Vorwort	**keine**
Verzeichnisse	Inhaltsverzeichnis	römisch, bei eins beginnend
	Abkürzungsverzeichnis	römisch, fortlaufend
	Abbildungsverzeichnis	römisch, fortlaufend
	Tabellenverzeichnis	römisch, fortlaufend

Diese arabische Seitenzählung wird bis zum Ende der Abhandlung beibehalten. Tabelle 2 illustriert die Auswirkungen der neu gewählten Nummerierung für alle restlichen Teile der wissenschaftlichen Arbeit.

Tabelle 2: Seitennummerierung des restlichen Dokuments

Komponente	Bestandteile	Seitennummerierung
Textteil	Kapitel eins	arabisch, bei eins beginnend
	weitere Kapitel	arabisch, fortlaufend
Anhang	alle Bestandteile	arabisch, fortlaufend
Verzeichnisse	Literatur	arabisch, fortlaufend
	Glossar	arabisch, fortlaufend
	Index	arabisch, fortlaufend
	Versicherung (eidesstattlich)	**keine**

3 Implementierung

Dieser Abschnitt vertieft Details, die im Zusammenhang mit der Umsetzung des Pakets fomsdt stehen. Dabei werden neben der eigentlichen Implementierung auch die Bereiche der Erstellung des Handbuchs und die Erzeugung einer zur Auslieferung geeigneten Version diskutiert. Doch vorab aber noch einige kurze Anmerkungen zu verschiedenen Schwierigkeiten und Problemen, die während der Umsetzung auftraten.

Basierend auf einer LaTeX-Präambel, die zum Zweck der Erstellung von Fallstudien und Seminararbeiten im Fachbereich Informatik der Fachhochschule für Oekonomie & Management in Essen erstellt wurde, begann im Sommer 2007 die Implementierungsphase des *Pakets zu Erstellung wissenschaftlicher Arbeiten*.

Diese Präambel beinhaltete bereits zu Implementierungsbeginn einige wesentliche Kommandos zur Textformatierung. Jedoch war relativ schnell abzusehen, dass sich eine simple Präambel schlecht bis ungenügend als allgemeine Vorlage für wissenschaftliche Arbeiten eignen würde. Dieser Umstand bedingte dann die Suche nach einer geeigneten und vernünftigen Strukturierung der einzelnen Bestandteile der ursprünglichen Präambel.

Nach umfangreichen Recherchen und einem Vergleich der verfügbaren Möglichkeiten wurde schlussendlich die Entscheidung getroffen, eine neue Dokumentenklasse zu erzeugen, die ihre funktionsorientierten Bestandteile in voneinander unabhängige Komponenten auslagert. Dieser Entschluss hatte jedoch weitere Anstrengungen zur Folge, um allen zukünftigen Anwendern das Umsetzungsresultat auch bereitstellen zu können. Hierdurch wurde dann die Generierung eines in LaTeX üblichen Installationsskriptes notwendig.

Die eventuellen Bedürfnisse zukünftiger Autoren standen ebenfalls im Mittelpunkt der planerischen Betrachtungen. Hierbei ergab sich, dass die Fähigkeiten der in der Basispräambel verfügbaren Kommandos stark auf spezifische Belange abgestimmt waren. Somit musste auch eine allgemeingültige Form der Bereitstellung von Kommandos zur Textformatierung gefunden werden. Dabei war zu berücksichtigen, dass die Konfiguration der einzelnen Kommandos mit Hilfe von optionalen und obligatorischen Parametern möglich sein sollte.

Da es sich bei LaTeX um eine der Hypertext Markup Language (HTML) vergleichbare Makrosprache handelt, brachte die Forderung nach verschiedenen optionalen und obligatorischen Parametern zwei grundlegende Schwierigkeiten hervor. Zum einen besteht innerhalb des LaTeX-Systems die Einschränkung, dass Kommandos nur maximal neun Parameter verwenden dürfen und zum anderen ist die Anzahl der möglichen optionalen Parameter auf einen Kann-Wert pro Kommando limitiert.

Die Beschränkung auf maximal neun Argumente je Kommando kann zwar mit Hilfe eines Tricks[4] durchbrochen werden, jedoch war dessen Anwendung für eine erfolgreiche Umsetzung nicht notwendig. Denn für das einzige davon betroffene Kommando wurde aus Gründen der besseren Übersichtlichkeit eine andere Alternative gewählt.

Optionale Kommandoparameter

Eine größere Herausforderung stellte die Bereitstellung mehrerer optionaler Kommandoparameter dar. Zur Lösung dieses Problems wurde das aus der objektorientierten Programmierung bekannte Verfahren der Methodenüberladung angewendet und an die Gegebenheiten des LaTeX-Systems angepasst.

Da leider eine genaue Adaption der Überladung von Kommandos in LaTeX nicht umsetzbar ist, wurde hierfür ein kleiner Trick angewendet. Dabei wird ein zweites, beinahe gleichnamiges Kommando erzeugt, das über eine äquivalente Liste von Parametern verfügt. Nur sind die Argumente des Zwillings so definiert, dass sie bei einem Aufruf angegeben werden müssen. Die Übergabe aller optionalen Parameter erfolgt dann in eckigen Klammern.

Die Aufgabe des Hauptkommandos, oder besser des Hauptmakros, besteht nun darin, alle durch den Anwender angegebenen Parameter in die benötigte Form zu überführen und mit diesen das Zwillingsmakro aufzurufen. Wurden durch den Anwender ein oder mehrere optionale Parameter ausgelassen, dann wird das Hauptmakro an seinen Zwilling stattdessen die hier definierten Standardwerte übergeben. Die Definition eines Makros mit zwei optionalen und einem Pflichtparameter zeigt das Beispiel in Quelltext 1.

Quelltext 1: Makro mit zwei optionalen Parametern (Beispiel)

```
% Makro für den Benutzer
\newcommand{\Macro}[1][A]{% A = Standard für Parameter 1!
  \@ifnextchar[%] % Zweiter optionaler Parameter vorhanden?
    {\@iiMacro[#1]}
    {\@iiMacro[#1][B]}    % B = Standard für Parameter 2!
}

% Internes Makro mit allen Parametern.
\def\@iiMacro[#1][#2]#3{  % Übergabe aller Parameter.
  Param1=#1; % Verwendung optionaler Parameter 1.
  Param2=#2; % Verwendung optionaler Parameter 2.
  Param3=#3} % Verwendung Pflichtparameter des Makros!
```

Zur Vermeidung eines direkten Durchgriffs auf das nur intern verwendete Makro wird diesem ein @-Zeichen beigefügt. Hierdurch wird sichergestellt, dass ein

[4] Der Trick zur Umgehung der limitierten Anzahl der Kommandoargumente kann in den *TeX Frequently Asked Questions on the Web* unter `http://www.tex.ac.uk/cgi-bin/texfaq2html?label=moren9` (letzter Zugriff: 2007-12-25) nachgeschlagen werden.

Anwender von außen nur das Hauptmakro aufrufen kann. Zusätzlich wird die Benennung des internen Makros durch ein Doppel-I eingeleitet. Dabei handelt es sich um einen in LaTeX üblichen Weg, dem Betrachter des Quellcodes die Anzahl der optionalen Parameter anzuzeigen.

In diesem Zusammenhang muss jedoch klargestellt werden, dass LaTeX keine lokalen Makros und auch keine privaten Variablen unterstützt. Zwar erzeugt die Verwendung des @-Zeichens beim Kompilieren eines Dokuments einen Fehler, doch kann dieser mit leichter Hand unterdrückt werden, wenn dem Aufruf eines damit gespicktem Makros das Kommando `\makeatletter` vorangestellt wird.

Optionale Umgebungsparameter

Lange Zeit war ungeklärt, ob in LaTeX überhaupt eine Möglichkeit besteht, auch so genannte Umgebungen mit optionalen Parameterlisten auszustatten. Dieses Problem konnte nach mehreren vergeblichen Versuchen und weiteren intensiven Recherchen schließlich gelöst werden.

Letztlich stellte sich heraus, dass der gesuchte Mechanismus bereits bei allen zuvor durchgeführten Makroimplementierungen erfolgreich angewendet wurde. Somit musste das dort verwendete Verfahren nur auf die Besonderheiten einer LaTeX-Umgebung angepasst werden.

Quelltext 2: Umgebung mit zwei optionalen Parametern (Beispiel)

```
% Umgebung für den Benutzer
\newenvironment{Environment}[1][A]{% A = Standard für Parameter 1!
  \@ifnextchar[%] % Zweiter optionaler Parameter vorhanden?
    {\begin{@iiEnvironment}[#1]}
    {\begin{@iiEnvironment}[#1][B]}% B = Standard für Parameter 2!
}{
  \end{@iiEnvironment}
}

% Interner Umgebungsbeginn mit allen Parametern.
\def\@iiEnvironment[#1][#2]#3{ % Übergabe aller Parameter.
  Param1=#1; % Verwendung optionaler Parameter 1.
  Param2=#2; % Verwendung optionaler Parameter 2.
  Param3=#3} % Verwendung Pflichtparameter der Umgebung!

% Internes Umgebungsende ohne Parameter.
\def\end@iiEnvironment{} %
```

Quelltext 2 zeigt das Grundgerüst zur Definition einer Umgebung mit zum Beispiel zwei optionalen und einem Pflichtparameter, dessen Funktionsweise mit der Parameterverarbeitung der zuvor diskutierten Makrodefinition identisch ist.

Beide zuvor gezeigten Methoden zur Verwendung mehrerer optionaler Argumente wurden während der Implementierung häufig angewendet. Sie bilden somit die Basis aller Makros und aller Umgebungen, die dem Endanwender durch das Paket fomsdt bereitgestellt werden.

3.1 Paketstruktur

Wie zuvor bereits angedeutet, war neben der eigentlichen Umsetzung auch die Frage nach einer geeigneten Paketstruktur zu klären. Eine Betrachtung dieses Teilbereichs der Implementierungen soll Gegenstand der nun anschließenden Ausführungen sein.

Insgesamt standen zur Strukturierung des Pakets fomsdt drei mögliche Lösungsansätze zur Auswahl, bei denen es sich im Einzelnen um die Bereitstellung einer einfachen Präambel, die Implementierung einer einfachen Stildatei sowie um die Definition einer völlig neuen Dokumentenklasse handelt.

Eine Analyse der Vor- und Nachteile aller zur Entscheidung stehenden Strukturvarianten ergab, dass die Verwendung einer eigenständigen Dokumentenklasse, die über zusätzliche und funktionsorientierte Stile verfügt, einem Endbenutzer die größtmögliche Flexibilität bieten würde. Daher fiel die Wahl der Paketstruktur auf eine Kombination aus

- Erzeugung einer neuartigen Dokumentenklasse und
- Bereitstellung weiterer Stildateien.

Dabei obliegt der Dokumentenklasse die Verarbeitung allgemeingültiger Grundeinstellungen und die verschiedenen Stildateien dienen der Behandlung spezieller Anwendungsfälle, die ihrer Art entsprechend verschiedenen Gruppen zugeordnet werden können.

Im weiteren Analyseverlauf konnte festgestellt werden, dass eine Unterscheidung zwischen textorientierten und gleitobjektorientierten Elementen sich hervorragend als Basis für eine Gruppierung eignen würde.

Weiterhin konnte eine sinnvolle Unterteilung der verschiedenen Bereiche einer wissenschaftlichen Arbeit in Elemente für den Beginn und Elemente für das Ende eines Dokuments getroffen werden.

Übrig blieben nun noch einige wenige Makros der Urpräambel, die keiner der anderen Gruppe zugeordnet werden konnten. Daher wurden diese Elemente in einer eigenen Gruppe „Vermischtes“ zusammengefasst.

Das Ergebnis der zuvor angestellten Betrachtungen hatte eine Aufteilung der jeweiligen Gruppen in verschiedene Dateien zur Folge. Diese Dateien sollen nun erst einmal kurz mit ihren Kernaufgaben vorgestellt werden.

`fomsdt.cls` Die Definition der Dokumentenklasse zur Erstellung wissenschaftlicher Arbeiten ist Kernaufgabe der Datei `fomsdt.cls`.

Nebenbei werden hierin auch verschiedene Einstellungen zum Blattrand, der Absatzgrundformatierung, der Formatierung von Kopf- und Fußzeilen und auch der Fußnotenformatierung vorgenommen. Weiterhin werden durch diese Datei alle Makros zur Konfiguration und Erzeugung des Titelblattes bereitgestellt. Die Makros zur Anpassung und Verwendung von Sperrvermerk und Eidesstattlicher Versicherung sind ebenfalls in der Datei `fomsdt.cls` hinterlegt.

`fomdss.sty` Die Bereitstellung der Makros zur Erzeugung der im Dokumentenbeginn vorzufindenden Verzeichnisse ist Aufgabe der Datei `fomdss.sty`. Dazu zählen das Inhaltsverzeichnis, das Abkürzungsverzeichnis, das Abbildungsverzeichnis und auch das Tabellenverzeichnis.

`fomdes.sty` Die Definition der Makros zur Erzeugung der Verzeichnisse am Ende eines Dokumentes obliegt der Datei `fomdes.sty`. Hierin befindet sich unter anderem das Marko zur Erstellung des in wissenschaftlichen Abhandlungen außerordentlich wichtigen Literaturverzeichnisses. Makros zur Generierung des Indexes, eines Glossars und auch des Verzeichnisses für Codelistings werden ebenfalls in dieser Datei definiert.

`fompar.sty` Speziell zur Formatierung von Absätzen und anderen Textpassagen gedachte Makros sind Bestandteil der Datei `fompar.sty`. Hierzu zählen beispielsweise Makros zur Erzeugung eingerückter Absätze, die über einen vorangestellten Bezeichner verfügen. Eine Umgebung zur beidseitigen Randeinrückung, die sich zum Beispiel sehr gut zur Hervorhebung von Zitaten eignet, ist ebenfalls in dieser Datei enthalten.

`fomfod.sty` Makros zur formatierten Einbindung von Abbildungen, Tabellen und Codelistings in wissenschaftliche Abhandlungen werden durch die Datei `fomfod.sty` bereitgestellt.

`fomisc.sty` Alle anderen, nicht kategorisierbaren Makros des Pakets fomsdt sind in der Datei `fomisc.sty` zusammengefasst. Beispielsweise wird hierin ein Makro zur normungskonformen Darstellung des Datums oder auch Makros zur Einbindung der speziellen Symbole für *Registered*, *Copyright* und *Trademark* definiert.

Ein unabhängig von einer Gruppierung nach Zugehörigkeit zu betrachtender Aspekt stellt die Manipulation des Indexes dar. Da dieser Teil des Pakets fomsdt eine Ausnahme in Bezug auf die Verwendung durch den Endanwender bedeutet, muss dennoch an dieser Stelle zumindest darauf eingegangen werden. Denn die zur Anpassung des Indexes notwendige Konfiguration muss den Regeln des LaTeX-Systems folgen und ist somit in einer separaten Datei auszulagern! Vom inhaltlichen Zusammenhang gehört die Indexkonfiguration jedoch zu den Elementen des Dokumentenendes.

`fomidx.ist` Die Datei `fomidx.ist` enthält verschiedene Konfigurationen für die ansprechendere Gestaltung des Indexes einer wissenschaftlichen Arbeit. Sie wird durch das Programm `makeindex` verwendet und ist daher nur indirekt zu verwenden. Zudem enthält diese Datei keine Makros im klassischen Sinne.

3.2 Paketelemente

Dieses Kapitel riskiert nun endlich einen Blick auf die Ergebnisse der Implementierung des Pakets fomsdt. Um einen Gesamtüberblick zu erhalten und das Zusammenspiel der jeweiligen Bestandteile des *Pakets zur Erstellung wissenschaftlicher Arbeiten* zu erarbeiten, wird sich hierfür auf eine abstrakte Betrachtung der einzelnen Komponenten beschränkt. Für das eingehendere Studium der einzelnen Quelltexte wird empfohlen, die im Anhang befindlichen vollständigen Sourcecodes einzusehen.

3.2.1 Dokumentenklasse

Basierend auf der LaTeX-eigenen Klasse article definiert die Datei `fomsdt.cls` die neue, eigenständige Dokumentenklasse des *Pakets zur Erstellung wissenschaftlicher Arbeiten*. Dabei meinen die einzelnen Buchstaben des Dateinamens nichts anderes als *FOM Scientific Document Template*, also Vorlage für wissenschaftliche Dokumente.

Neben einer Bereitstellung der Dokumentenklasse hat die Datei `fomsdt.cls` noch die Aufgabe, die verschiedenen Grundeinstellungen für wissenschaftliche Abhandlungen vorzunehmen. Dazu zählen unter anderem das Einstellen der Schriftgröße auf zwölf Punkt und die vollständige Verwendung des beschreibbaren Bereichs eines Blattes der Größe DIN A4.

Das Laden verschiedener Basispakete zur Unterstützung spezieller Belange der deutschen Sprache gehört ebenso zu den Aufgaben der Datei `fomsdt.cls` wie

auch die Konfiguration der Kopf- und Fußzeilen. Verschiedene Anpassungen für Fußnoten werden ebenfalls hierin vorgenommen.

Ein weiteres Merkmal der Datei `fomsdt.cls` ist die Festlegung der zu Grunde liegenden Absatzformatierung. Hierunter fallen unter anderem die Justage des Abstands zwischen aufeinander folgende Absätze und auch die Unterdrückung des Einzugs der ersten Absatzzeile. Weitere Einstellungen dienen der Verhinderung einzelner Zeilen auf vorherigen und folgenden Textseiten, die in der Sprache der Schriftsetzer als Schusterjungen und Hurenkinder bezeichnet werden.

Diese Einstellungen sind allesamt von grundlegender Natur und im deutschen Sprachraum allgemein gängige Randbedingen zur Dokumentenerstellung. Daher bietet die Dokumentenklasse des Pakets fomsdt noch weitere Komponenten, die speziell auf die Erstellung wissenschaftlicher Ausarbeitungen zugeschnitten sind. Diese Komponenten sollen nun näher betrachtet werden.

Klassenoptionen

Die Dokumentenklasse selbst verfügt über eigene nützliche Optionen zur Konfiguration des Layouts der gesamten Arbeit, die nun der Reihe nach vorgestellt werden sollen.

coloredlinks

Die Option `coloredlinks` forciert die Verwendung farbiger Hyperlinks zur Hervorhebung von Querverweisen innerhalb eines Dokuments, da standardmäßig alle Querverweise in Schwarz dargestellt werden. Möchte ein Autor die Querverweise seines Dokuments nun farbig darstellen, dann kann er dies durch Angabe dieser Option erreichen. In diesem Fall wird ein dunkles Blau zur Hervorhebung der Querverweise verwendet.

noheadrule

Die Option `noheadrule` stellt die Strichstärke der Trennlinie zwischen der Kopfzeile und dem restlichen Text auf den Wert Null ein, da diese Linienstärke vom Paket fomsdt per Grundeinstellung auf einen Wert von $\frac{1}{2}$ Punkt voreingestellt wird. Hierdurch wird ein Autor befähigt, die optische Trennung zwischen Seitenkopf und Textteil zu unterdrücken.

nochapter

Die Option `nochapter` unterdrückt die Anzeige der Überschrift des Hauptkapitels in der Kopfzeile eines Dokuments. Somit muss ein Autor diese Kapitelüberschrift explizit ausschalten, wenn dies von ihm gewünscht wird.

Werden in den Klassenladeoptionen die Optionen `noheadrule` und `nochapter` miteinander kombiniert, dann kann hierdurch die Verwendung der Kopfzeile gänzlich unterdrückt werden!

moderntitle

Die Option `moderntitle` verwendet ein modern anmutendes Layout für die Titelseite der wissenschaftlichen Arbeit.

classictitle

Die Option `classictitle` verwendet ein klassisch anmutendes Layout für die Titelseite der wissenschaftlichen Arbeit.

Wird in den Ladeoptionen der Klasse als Stil der Titelseite weder `moderntitle` noch `classictitle` angegeben, dann können zur Erzeugung des Titelblattes weiterhin die LaTeX-eigenen Makros `\title`, `\author` et cetera verwendet werden. In diesem Fall wird vom Paket fomsdt jedoch eine Warnmeldung erzeugt, die den Benutzer darauf aufmerksam machen soll, dass er eventuell die Informationen zur Erzeugung der Titelseite vergessen hat.

Werden hingegen beide Stiloptionen gemeinsam verwendet, die Reihenfolge ist dabei unerheblich, dann wird als Standard für die Titelseite der klassisch anmutende Stil verwendet.

lockflag

Die Option `lockflag` schaltet die Verwendung des Sperrvermerks ein. Da ein Sperrvermerk in der Regel nur bei Diplomarbeiten anzutreffen ist, muss dieser durch den Autor ausdrücklich aktiviert werden!

affidavit

Die Option `affidavit` schaltet die Verwendung der Eidesstattlichen Versicherung ein. Da die ehrenwörtliche Erklärung grundsätzlich nur in Diplomarbeiten verwendet wird, ist diese ebenfalls zuvor durch den Autor zu aktivieren!

Titelseite

Das mit Abstand quellcode-intensivste Einzelmakro innerhalb des gesamten Pakets fomsdt ist das Makro zur Erstellung der Titelseite. Dieser Umstand ist aber nicht auf eine hohe inhärente Komplexität zurückzuführen, wie man eventuell vermuten könnte, sondern begründet sich dadurch, dass vielerlei Anstrengungen notwendig sind, um einem Benutzer wenigsten zwei unterschiedliche Varianten eines Abhandlungstitels zu Verfügung stellen zu können.

Zudem bestehen diverse Abhängigkeiten zu anderen Makros, die einer Reduzierung der bereits vorhandenen Komplexität nicht gerade zuträglich sind. Die Aufschlüsselung dieser Vernetzungen soll Gegenstand dieses Teilabschnittes sein.

Üblicherweise werden in LaTeX die Titelseiten der Dokumente mit Hilfe des vordefinierten Makros `\maketitle` erstellt. Dieses Makro wird im Paket fomsdt ebenfalls zu Erstellung des Titels verwendet. Zu diesem Zweck wird es durch

die Datei `fomsdt.cls` neu definiert. Dabei wird eine Vorauswahl anhand der angegebenen Klassenoptionen `moderntitle` und `classictitle` getroffen.

<table>
<tr><td colspan="3">Klassenladeoption für die Titelseite?</td></tr>
<tr><td>Moderner Titel</td><td>Klassischer Titel</td><td>keine</td></tr>
<tr><td>• Titelblatt erzeugen
• Randbereich zurücksetzen
• Texthöhe und -breite speichern
• Texthöhe und -breite neu einstellen
• Linke Textbox (rotiert) erzeugen
• Text in linker Textbox ausgeben
• Verwende <code>\General{}</code>
• Rechte Textbox erzeugen
• Text in rechter Textbox rechtsbündig
• Text in rechter Textbox ausgeben
• Verwende <code>\Institute{}</code>
• Verwende <code>\Course{}</code>
• Verwende <code>\Semester{}</code>
• Verwende <code>\Subject{}</code>
• Verwende <code>\Title{}</code>
• Untere Textbox erzeugen
• Text in unterer Textbox ausgeben
• Verwende <code>\Professor{}</code>
• Verwende <code>\Student{}</code>
• Verwende <code>\Date{}</code>
• Alte Texthöhe und -breite zurücksetzen</td><td>• Titelblatt erzeugen
• Obere Textbox erzeugen
• Text in oberer Textbox zentriert
• Text in oberer Textbox ausgeben
• Verwende <code>\Institute{}</code>
• Verwende <code>\Course{}</code>
• Verwende <code>\Semester{}</code>
• Verwende <code>\Subject{}</code>
• Verwende <code>\Title{}</code>
• Untere Textbox erzeugen
• Text in unterer Textbox ausgeben
• Verwende <code>\Professor{}</code>
• Verwende <code>\Student{}</code>
• Verwende <code>\Date{}</code></td><td rowspan="4">• Warnmeldung ausgeben!</td></tr>
<tr><td colspan="2">Sperrvermerk ausgeben?</td></tr>
<tr><td>JA</td><td>NEIN</td></tr>
<tr><td>• Neue Seite erzeugen
• Sperrvermerk ausgeben</td><td></td></tr>
</table>

Abbildung 9: Funktionsweise des neu definierten Makros `\maketitle`

Wurde durch Angabe der Option `lockflag` in den Klassenladeoptionen die Verwendung des Sperrvermerks aktiviert, dann werden zu dessen Erzeugung zusätzliche Angaben erforderlich!

Da eine detaillierte Betrachtung des Sperrvermerks im hieran anschließenden Teilabschnitt erfolgt, sollte an dieser Stelle zur Diskussion des Erstellverfahrens der Titelseite die Information „Der Sperrvermerk wird oder wird nicht verwendet“ hinreichend genau sein.

Nachdem durch den Benutzer das Layout der Titelseite festgelegt wurde, hat er die Aufgabe, weitere Angaben zur Titelseite vorzunehmen. Dies geschieht mit Hilfe der in Abbildung 9 angedeuteten Makros, die Gegenstand der nun folgenden Betrachtungen sein sollen. Dabei reicht ein flüchtiger Blick völlig aus, da alle gezeigten Makros ihre Daten nur speichern und nicht verarbeiten.

\General

Das Makro `\General` legt die Ausprägung der wissenschaftlichen Ausarbeitung fest. Hier könnte der Benutzer zum Beispiel „Diplomarbeit“ oder auch „Studienarbeit“ angeben. Dabei handelt es sich um einen optionalen Parameter, dessen Standardwert vom Paket auf *Hausarbeit* eingestellt wird.

\Institute

Das Makro `\Institute` legt den Namen der Hochschule fest, an der die wissenschaftliche Ausarbeitung geschrieben wurde. Es handelt sich hierbei ebenfalls um einen optionalen Parameter, dessen Standardwert auf *FOM – Fachhochschule für Oekonomie & Management Essen* voreingestellt ist.

\Course

Das Makro `\Course` definiert den Studiengang, für den die wissenschaftliche Arbeit erstellt wurde. Auch hier handelt es sich um einen optionalen Parameter, dessen Standardwert *Berufsbegleitender Studiengang* ist.

\Semester

Das Makro `\Semester` definiert das Semester, in dem die wissenschaftliche Ausarbeitung erstellt wurde. Dieser Wert ist optional und verfügt daher über keinen vordefinierten Standardwert.

\Date

Das Makro `\Date` legt das Datum der wissenschaftlichen Ausarbeitung fest. Hierbei handelt es sich auch um einen optionalen Parameter, dessen Standardwert *Essen, den \today* ist.

\Subject

Das Pflichtmakro `\Subject` wird verwendet, um das Studienfach beziehungsweise das Abhandlungsthema anzugeben. Bei Diplomarbeiten kann dieses Kommando beispielsweise hervorragend zur Angabe von „Diplomarbeit zum Thema“ verwendet werden.

\Title

Das Pflichtmakro `\Title` definiert den Titel der wissenschaftlichen Arbeit. Dem ist sicherlich nichts weiter hinzuzufügen.

\Professor

Das Makro `\Professor` definiert den Namen des Betreuers der wissenschaftlichen Arbeit. Es verfügt, im Gegensatz zu allen vorherigen Makros, sowohl über

einen optionalen als auch über einen Pflichtparameter. Dabei ist der Standardwert des optionalen Arguments ⟨*Bezeichner*⟩ auf *Betreuer:* eingestellt.

`\Student`

Das Makro `\Student` definiert den Namen des Autors der wissenschaftlichen Ausarbeitung. Es verfügt ebenfalls über einen optionalen und einen Pflichtparameter. Der Standardwert des optionalen Arguments ⟨*Bezeichner*⟩ wird hierbei mit dem Wert *Autor:* initialisiert.

Auch wenn die durchgeführten Implementierungen zur Erzeugung der Titelseite relativ umfangreich waren, ist dennoch die Benutzug der hierfür notwendigen Makros sehr leicht möglich, wie Quelltext 3 beispielhaft demonstriert.

Quelltext 3: Konfiguration und Erzeugung der Titelseite

```
% Klasse laden mit modernem Titel
\documentclass[moderntitle]{fomsdt}
\begin{document}
% Titelkonfiguration
\General   {Bezeichnung der Ausarbeitung}
\Date      {Stadt, den \sl Abgabedatum}
\Institute{Name der Hochschule}
\Course    {Bezeichnung des Fachs}
\Semester {Semesterangabe}
\Subject   {Thema der Ausarbeitung}
\Title     {Titel der Ausarbeitung}
\Professor[Betreuer:]{Titel Vorname Nachname}
\Student   [Autor:]{Vorname Nachname (Matrikelnummer)}
% Titelerzeugung
\maketitle
\end{document}
```

Das in Quelltext 3 gezeigte Codefragment beinhaltet ein vollständiges und mit LaTeX kompilierbares Dokument, das über die moderne Variante des Titelblattes verfügt. Dabei übernimmt das im Paket fomsdt überschriebene Makro `\maketitle` alle notwendigen Aufgaben zu dessen Erzeugung.

Da in den optionalen Parametern des Makros `\documentclass` des gezeigten Beispiels keine Angabe zur Verwendung des Sperrvermerks erfolgte, also die Klassenoption `lockflag` nicht angegeben wurde, besteht ein hieraus generiertes Dokument einzig und allein aus der Titelseite.

Soll anstatt des modernen Titels doch lieber die klassische Variante zum Einsatz kommen, dann ist im Quelltext lediglich die Option `moderntitle` durch `classictitle` zu ersetzen und das Dokument erneut zu kompilieren.

Ein Beispiel für beide Variationen der Titelseite, die durch das Paket fomsdt unterstützt werden, zeigt Abbildung 10.

(a) Klassenoption `moderntitle`

(b) Klassenoption `classictitle`

Abbildung 10: Beispiele zur Variation des Titelblattes im Paket fomsdt

Sperrvermerk

Da per Definition der Sperrvermerk auf der zweiten Seite der wissenschaftlichen Arbeit vorhanden sein soll, kann dieser dem Abhandlungsvorspann und somit der Titelseite zugerechnet werden. Aus diesem Grund wurde die Aufgabe des Makros `\maketitle` entsprechend erweitert, sodass die Erzeugung des Titels und die Anzeige des Sperrvermerks Hand in Hand gehen können. Dieses Verhalten deutete bereits Abbildung 9 an.

Allgemein betrachtet verfügen Sperrvermerke über einen Inhalt, der normalerweise nicht angepasst werden muss. Jedoch konnten während der Umsetzungsphase verschiedene Parameter ausgemacht werden, die eine Flexibilisierung des vom Paket fomsdt bereitgestellten Sperrvermerks als sinnvoll erscheinen ließen.

Wie bereits weiter oben erwähnt, wird zur Aktivierung des Sperrvermerks ausschließlich die Angabe der Klassenoption `lockflag` benötigt, da dieser zusammen mit dem Abhandlungstitel automatisch generiert wird. Jedoch sind die durch den Titel festgelegten Informationen zu dessen Erzeugung unzureichend! Daher sind durch den Anwender weitere Details bereitzustellen, die nun etwas genauer unter die Lupe genommen werden sollen.

\LockflagContent

Mit Hilfe des Makros `\LockflagContent` ist es möglich, den grundlegenden Inhalt des Sperrvermerks zu steuern. Hierbei bestimmt der Pflichtparameter `{`⟨*Inhalt*⟩`}` den zu selektierenden Basistext des Sperrvermerks wie folgt:

- `std` Wird `\LockflagContent{std}` verwendet, dann enthält der Sperrvermerk einen Standardtext. Dieser Standardtext wird bereits zum Zeitpunkt des Ladens der Dokumentenklasse voreingestellt.
- `fom` Wird `\LockflagContent{fom}` verwendet, dann enthält der Sperrvermerk einen Text, der auf die besonderen Belange der Fachhochschule für Oekonomie & Management abgestimmt ist.
- `...` Wird weder `std` noch `fom` als Parameter angegeben, dann enthält der Sperrvermerk den in den Punkten angegebenen und somit frei konfigurierbaren Text.

Ebenfalls für das Makro `\LockflagContent` verfügbar ist der optionale Parameter `[`⟨*Signatur*⟩`]`. Dieser Parameter erlaubt die Aktivierung der Signatur des Sperrvermerks folgendermaßen:

- `sig` Wird als optionaler Parameter `sig` verwendet, dann enthält der Sperrvermerk eine aus den Feldern *(Ort, Datum)* und *(Eigenhändige Unterschrift)* bestehende Signatur.

Zur Anpassung des linken Signaturteils, also zur Manipulation des Inhalts des Feldes *(Ort, Datum)*, kann der Anwender das Makro `\LockflagSigLeft{`⟨*linker Signaturteil*⟩`}` anwenden. Die Änderung des rechten Teils der Signatur, also die Anpassung des Inhalts von *(Eigenhändige Unterschrift)* ist ebenfalls möglich. Zu diesem Zweck kann der Benutzer das Makro `\LockflagSigRight{`⟨*rechter Signaturteil*⟩`}` verwenden.

\LockflagCompany

Da durch einen Sperrvermerk die geheimen Informationen eines bestimmten Unternehmens geschützt werden sollen, darf der entsprechende Firmenname auch nicht im Sperrvermerk fehlen. Zur Bekanntgabe dieses Namens definiert das Paket fomsdt das Makro `\LockflagCompany`.

Im Zusammenhang mit dem Sperrvermerk des Pakets fomsdt stehen einem Anwender noch weitere, jedoch weniger wichtige Makros zur Verfügung, auf die hier nur kurz eingegangen werden soll.

\LockflagDate

Das Makro \LockflagDate kann verwendet werden, um den realen Ort sowie das Datum der Erstellung innerhalb der Signatur des Sperrvermerks zu setzen. Somit ist der Anwender frei vom Zwang, diese Information handschriftlich in das Feld *(Ort, Datum)* eintragen zu müssen.

\LockflagHeader

Zur Änderung der Standardüberschrift des Sperrvermerks steht dem Anwender das Makro \LockflagHeader zur Verfügung. Diese Makro ist beispielsweise dann besonders hilfreich, wenn anstelle der Überschrift *Sperrvermerk* ein gänzlich anderer Text verwendet werden soll.

Basierend auf dem bereis in Quelltext 3 verwendeten Codefragment sollen zur Demonstration der Möglichkeiten des Sperrvermerks hier nun noch zwei Beispiele gezeigt werden.

Beispiel für den Standardsperrvermerk ohne Signatur

Das Beispiel in Quelltext 4 zeigt den Code, der zur Erstellung eines Standardsperrvermerks notwendig ist. Dabei ist zu beachten, dass der generierte Sperrvermerk ohne zusätzlicher Signatur erzeugt wird.

Quelltext 4: Erzeugung des Standardsperrvermerks ohne Signatur

```
% Klasse laden mit modernem Titel und
% Aktivierung des Standardsperrvermerks
\documentclass[moderntitle,lockflag]{fomsdt}
\begin{document}
% Titelkonfiguration
\General   {\it Bezeichnung der Ausarbeitung}
\Date      {Stadt, den \sl Abgabedatum}
\Institute{Name der Hochschule}
\Course    {Bezeichnung des Fachs}
\Semester {Semesterangabe}
\Subject   {Thema der Ausarbeitung}
\Title     {Titel der Ausarbeitung}
\Professor[Betreuer:]{Titel Vorname Nachname}
\Student   [Autor:]{Vorname Nachname (Matrikelnummer)}
% Sperrvermerkkonfiguration
\LockflagCompany{\it Name des Unternehmens}
% Titelerzeugung
\maketitle
\end{document}
```

Wurde das zugehörige Dokument kompiliert, dann stellt sich das Ergebnis des im Quelltext 4 verwendenden Codes, wie in Abbildung 11 gezeigt, dar. Zu beachten ist hierbei, dass sich der erzeugte Sperrvermerk auf der zweiten Seite des Ausgabedokuments befindet, da die erste Seite dem Titelblatt vorbehalten ist.

Sperrvermerk

Die vorliegende *Bezeichnung der Ausarbeitung* beinhaltet interne vertrauliche Informationen der Firma *Name des Unternehmens*. Die Weitergabe des Inhalts der *Bezeichnung der Ausarbeitung* im Gesamten oder in Teilen sowie das Anfertigen von Kopien oder Abschriften – auch in digitaler Form – sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Genehmigung der Firma *Name des Unternehmens*.

Abbildung 11: Ausgabe des Standardsperrvermerks ohne Signatur

Beispiel für den FOM-spezifischen Sperrvermerk mit Signatur

Das Beispiel in Quelltext 5 zeigt den Code, der zur Erstellung des speziell für die Fachhochschule für Oekonomie & Management (FOM) zu verwendenden Sperrvermerks notwendig ist. Dabei ist zu beachten, dass der generierte Sperrvermerk mit zusätzlicher Signatur erzeugt wird. Das angegebene Signaturdatum wird ebenfalls verwendet.

Quelltext 5: Erzeugung des Sperrvermerks der FOM mit Signatur

```
% Klasse laden mit modernem Titel und
% Aktivierung des Standardsperrvermerks
\documentclass[moderntitle,lockflag]{fomsdt}
\begin{document}
% Titelkonfiguration
\General   {\it Bezeichnung der Ausarbeitung}
\Date      {Stadt, den \sl Abgabedatum}
\Institute{Name der Hochschule}
\Course    {Bezeichnung des Fachs}
\Semester {Semesterangabe}
\Subject   {Thema der Ausarbeitung}
\Title     {\it Titel der Ausarbeitung}
\Professor[Betreuer:]{Titel Vorname Nachname}
\Student   [Autor:]{Vorname Nachname (Matrikelnummer)}
% Sperrvermerkkonfiguration
\LockflagCompany{\it Name des Unternehmens}
\LockflagDate{\it Stadt, Datum}
\LockflagContent[sig]{fom}
% Titelerzeugung
\maketitle
\end{document}
```

Nach erfolgter Kompilierung des zugehörigen Dokuments stellt sich das Ergebnis des im Quelltext 5 verwendeten Codes, wie in Abbildung 12 gezeigt, dar. Auch hier gilt es zu beachten, dass sich der erzeugte Sperrvermerk auf der zweiten Seite des Ausgabedokuments befindet.

Sperrvermerk

Die vorliegende *Bezeichnung der Ausarbeitung* mit dem Titel *Titel der Ausarbeitung* enthält unternehmensinterne Daten der Firma *Name des Unternehmens*. Daher ist sie nur zur Vorlage bei der FOM sowie den Begutachtern der Arbeit bestimmt. Für die Öffentlichkeit und dritte Personen darf sie nicht zugänglich sein.

Stadt, Datum
(Ort, Datum) (Eigenhändige Unterschrift)

Abbildung 12: Ausgabe des Sperrvermerks der FOM mit Signatur

Abschließend gilt noch festzuhalten, dass die Verwendung des Sperrvermerks einige Abhängigkeiten zu anderen Makros des Pakets fomsdt verursacht! Diese Abhängigkeiten sollen nun näher aufgeschlüsselt werden.

- Wird das Makro `\LockflagContent` mit dem Argument `fom` verwendet, dann werden durch den für die Fachhochschule für Oekonomie & Management spezifischen Sperrvermerk die Daten der Makros `\General`, `\Title` und `\LockflagCompany` verwendet!
- Wird hingegen das Makro `\LockflagContent` mit dem Argument `std` verwendet, dann werden durch den Standardsperrvermerk die Daten der Makros `\General` und `\LockflagCompany` verwendet!
- Wird als Parameter des Makros `\LockflagContent` ein freier Text angegeben, dann besteht keine Abhängigkeit zu anderen Makros!
- Sollte durch die Verwendung des optionalen Parameters `sig` im Makro `\LockflagContent` die Signatur des Sperrvermerks aktiviert worden sein, dann werden in diesem Fall die Daten des Makros `\LockflagDate` verwendet!

Ehrenwörtliche Erklärung

Im Gegensatz zum Sperrvermerk ist die ehrenwörtliche Erklärung oder auch Eidesstattlichen Versicherung auf der letzten Seite einer wissenschaftlichen Ab-

handlung zu platzieren. Dadurch wird ihr Charakter nochmals betont. Denn sie gilt als Versprechen des Autors, alle Inhalte selbstständig erarbeitet und nicht von anderen abgeschrieben zu haben. Dies wird dann auch durch die eigenhändige Unterschrift des Autors bezeugt.

Da es sich bei der Eidesstattlichen Versicherung um einen Pflichtbestandteil für Diplomarbeiten handelt, wird diese vom Paket fomsdt ebenfalls in vorkonfigurierter Form zur Verfügung gestellt. Dabei orientiert sich deren Verwendung sehr stark an der Verwendung des Sperrvermerks. Somit muss auch die ehrenwörtliche Erklärung explizit aktiviert werden. Hierfür ist dann, ähnlich wie beim Sperrvermerk, die Angabe der Klassenladeoption `affidavit` erforderlich.

Wurde die Verwendung der ehrenwörtlichen Erklärung durch den Benutzer angestoßen, dann wird diese während der Erstellung des Dokuments automatisch erzeugt. Zur Sicherstellung, dass sich das Autorenversprechen auch tatsächlich auf der letzten Abhandlungsseite befindet, wird in der Datei `fomsdt.cls` das LaTeX-eigene Makro `\AtEndDocument` verwendet.

Wie schon angedeutet, bestehen zwischen Sperrvermerk und Eidesstattlicher Versicherung sehr viele Parallelen. Diese Gemeinsamkeiten spiegeln sich sowohl im Umfang als auch in der Benennung der zur Konfiguration verfügbaren Kommandos wieder. Daher wird es als sinnvoll erachtet, an dieser Stelle nicht allzu intensiv auf sämtliche Details der Erzeugung der ehrenwörtlichen Erklärung einzugehen, sondern besser die Unterschiede herauszuarbeiten.

Unüblich und somit überflüssig ist die Verwendung eines Firmennamens innerhalb einer Eidesstattlichen Versicherung, da der Autor ja hierdurch seine eigene Ehrlichkeit verbriefen soll. Daher wird vom Paket fomsdt auch kein dem Makro `\LockflagCompany` äquivalentes Makro für die Einbeziehung einer Firma in die ehrenwörtliche Erklärung angeboten.

Das Makro `\AffidavitContent` dient, analog zum Makro `\LockflagContent`, der Festlegung des Inhalts der Eidesstattlichen Versicherung. Es verwendet ebenfalls sowohl den Pflichtparameter {⟨*Inhalt*⟩} als auch den optionalen Parameter [⟨*Signatur*⟩]. Beide Parameter verarbeiten die Benutzereingaben auf die gleiche Art und Weise, wie dies beim Makro `\LockflagContent` der Fall ist.

Dennoch gibt es hier einen Unterschied! Denn die vorkonfigurierte Eidesstattliche Versicherung verfügt nicht über das Argument `fom`. Dieses Argument ist auch überflüssig, da hierbei eine Unterscheidung zwischen einer standardisierten Eidesstattlichen Versicherung und einer speziell für die Fachhochschule für Oekonomie & Management erstellten ehrenwörtlichen Erklärung unnötig ist.

Im Kern bedeutet das nichts anderes als dass die durch das Paket fomsdt verwendete Standarderklärung in ihrem Wortlaut mit der durch die Fachhochschule für Oekonomie & Management bevorzugten ehrenwörtlichen Erklärung

übereinstimmt! Einzige Ausnahme ist, dass an zwei Stellen der Begriff „Arbeit“ der originären Version durch den Inhalt des Makros `\General` ersetzt wird.

Daher ist ein Abweichen von der standardisierten Erklärung nicht notwendigerweise gegeben, da diese bereits alle wichtigen Aspekte einer Eidesstattlichen Versicherung enthält. Wünscht ein Anwender dennoch eine eigene Variante, dann bleibt ihm immer noch die Verwendung des frei konfigurierbaren Textes im Makro `\AffidavitContent`.

Die anderen zur Konifiguration der Eidesstattlichen Versicherung verfügbaren Kommandos `\AffidavitDate`, `\AffidavitHeader`, `\AffidavitSigLeft` und `\AffidavitSigRight` verarbeiten ihre Daten auf die exakt gleiche Weise, wie dies bereits ihre namensverwandten Markos für den Sperrvermerk vollführen.

Wie auch schon im vorherigen Teilabschnitt, hier nun ein Beispiel zur Veranschaulichung der Verwendung der Eidesstattlichen Versicherung, basierend auf dem bereits in Quelltext 3 verwendeten Codefragment.

Beispiel für die ehrenwörtliche Erklärung ohne Signatur

Das Beispiel in Quelltext 6 zeigt den Quelltext, der für die Erstellung einer Eidesstattlichen Versicherung notwendig ist. Dabei ist zu beachten, dass die generierte Erklärung mit zusätzlicher Signatur erzeugt wird. Das angegebene Signaturdatum wird ebenfalls verwendet. Zudem wird in diesem Beispiel noch die auszugebende Überschrift geändert.

Quelltext 6: Erzeugung der ehrenwörtlichen Erklärung mit Signatur

```
% Klasse laden mit modernem Titel und
% Aktivierung der ehrenwörtlichen Erklärung
\documentclass[moderntitle,affidavit]{fomsdt}
\begin{document}
% Titelkonfiguration
\General   {\it Bezeichnung der Ausarbeitung}
\Date      {Stadt, den \sl Abgabedatum}
\Institute{Name der Hochschule}
\Course    {Bezeichnung des Fachs}
\Semester {Semesterangabe}
\Subject   {Thema der Ausarbeitung}
\Title     {Titel der Ausarbeitung}
\Professor[Betreuer:]{Titel Vorname Nachname}
\Student   [Autor:]{Vorname Nachname (Matrikelnummer)}
% Konfiguration der ehrenwörtlichen Erklärung
\AffidavitHeader{\it Versicherung (eidesstattlich)}
\AffidavitDate{\it Stadt, Datum}
\AffidavitContent[sig]{std}
% Titelerzeugung
\maketitle
% Automatische Erzeugung der ehrenwörtlichen Erklärung
\end{document}
```

Die Kompilierung des zum Codebeispiel aus Quelltext 6 gehörenden Dokuments ergibt dann das in Abbildung 13 dargestellte Ergebnis.

Versicherung (eidesstattlich)

Hiermit versichere ich, dass die vorliegende *Bezeichnung der Ausarbeitung* von mir selbstständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere, dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Weiterhin erkläre ich, dass die *Bezeichnung der Ausarbeitung* in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat.

Stadt, Datum
(Ort, Datum) (Eigenhändige Unterschrift)

Abbildung 13: Ausgabe der ehrenwörtlichen Erklärung mit Signatur

Bleibt noch anzumerken, dass die Verwendung der ehrenwörtlichen Erklärung verschiedene Abhängigkeiten zu anderen Makros des Pakets fomsdt verursacht. Diese Abhängigkeiten werden hier nun kurz zusammengefasst.

- Wird das Makro `\AffidavitContent` zusammen mit dem Argument `std` verwendet, dann wird von der Eidesstattlichen Versicherung der Inhalt des Makros `\General` verwendet!
- Wird als Parameter des Makros `\AffidavitContent` ein freier Text angegeben, dann besteht keine Abhängigkeit zu anderen Makros!
- Sollte durch die Verwendung des optionalen Parameters `sig` im Makro `\AffidavitContent` die Signatur der Eidesstattlichen Versicherung aktiviert worden sein, dann werden in diesem Fall die Daten des Makros `\AffidavitDate` verwendet!

Spezielle Makros für PDF-Dateien

Das Paket fomsdt bietet den Benutzern spezielle Makros, die zusammen mit der Erstellung von Dateien im PDF-Format sehr nützlich sind. Dabei besteht deren Aufgabe darin, ausgewählte Informationen der PDF-Ausgabedatei so zu verändern, dass sie in den PDF-eigenen Eigenschaften hinterlegt werden. Dies

funktioniert aber nur dann, wenn zur Kompilierung des Dokuments auch das Programm `pdfLaTeX` verwendet wurde. Anderenfalls werden eventuell gesetzte Informationen ignoriert.

\PDFTitle

Das Makro `\PDFTitle` ändert die Eigenschaft ⟨*Titel*⟩ der PDF-Ausgabedatei. Ein Standardwert wird von diesem Makro nicht verwendet.

\PDFAuthor

Das Makro `\PDFAuthor` ändert die Eigenschaft ⟨*Autor*⟩ der PDF-Ausgabedatei. Ein Standardwert wird von diesem Makro nicht verwendet.

\PDFSubject

Das Makro `\PDFSubject` ändert die Eigenschaft ⟨*Thema*⟩ der PDF-Ausgabedatei. Ein Standardwert wird von diesem Makro nicht verwendet.

\PDFProducer

Das Makro `\PDFProducer` ändert die Eigenschaft ⟨*Erzeuger*⟩ der PDF-Ausgabedatei und verwendet *PDFLaTeX* als Standardwert.

Wie leicht festzustellen ist, existieren bei den hier vorgestellten Makros gewisse Analogien zu einigen für das Setup der Titelseite verfügbaren Makros. Manch einer wird sich sicherlich die Frage stellen, warum nicht bereits die vorhandenen Informationen für die Speicherung in den PDF-Eigenschaften verwendet wurden. Die Begründung ist recht einleuchtend.

Da grundsätzlich die Möglichkeit besteht, zusätzliche Anweisungen zur Formatierung, sprich weitere Makros, in den Makros zur Titelerzeugung anzugeben, können die so gesetzten Informationen nicht vollständig in die PDF-eigenen Eigenschaften der Ausgabedatei übernommen werden. Somit kann es im einfachsten Fall zu hieroglyphenartigen Zeichenketten in den PDF-Eigenschaften kommen. Im schlimmsten Fall wird sogar der Kompiliervorgang mit einer Fehlermeldung abgebrochen. Aus diesem Grund existieren parallele Makros, die unter Umständen auch redundante Informationen enthalten können.

3.2.2 Dokumentenbeginn

Der Beginn einer Abhandlung mit wissenschaftlichem Hintergrund ist, wie bereits mehrfach angesprochen, gekennzeichnet durch die Existenz typischer Verzeichnisse. Die Bereitstellung von Makros zur Erzeugung und Verwaltung dieser Verzeichnisse ist Aufgabe der Datei `fomdss.sty`. Dabei bedeuten die Buchstaben des Dateinamens im Einzelnen *FOM Document Start Section*.

Optionen des Dokumentbeginns

Auch der Dokumentenbeginn verfügt über eigene nützliche Optionen zur Konfiguration des Layouts der einzelnen Verzeichnisse, die hier der Reihe nach einmal vorgestellt werden sollen.

toctotoc

Die Option `toctotoc` hat die Aufgabe, die Überschrift des Inhaltsverzeichnisses ebenfalls mit in das Inhaltsverzeichnis aufzunehmen. Einige werden sich sicherlich fragen, was diese Option überhaupt für einen Nutzen hat.

Stimmt, diese Option hat keinen Nutzen und dennoch ist sie notwendig! Denn leider existieren so manche Richtlinien, in denen tatsächlich befürwortet wird, die Überschrift des Inhaltsverzeichnisses ebenfalls im Inhaltsverzeichnis aufzuführen; mit Angabe der Seitenzahl versteht sich [13, 15].

abbreviation

Die Option `abbreviation` aktiviert die Verwendung des Abkürzungsverzeichnisses. Die Einführung dieser Option wurde notwendig, da die Erstellung des Abkürzungsverzeichnisses auf Basis des Pakets nomencl erfolgt. Leider gehört dieses Paket noch nicht zum Standardumfang der verschiedenen LaTeX-Distributionen und muss somit eventuell nachträglich installiert werden.

Um von vornherein Kompilierungsfehler zu vermeiden, die aufgrund des Fehlens des Pakets nomencl entstehen können, muss die Verwendung des Abkürzungsverzeichnisses vom Anwender ausdrücklich gewollt werden!

Außerdem kommt hinzu, dass ein mit nomencl erstelltes Abkürzungsverzeichnis nur mit Hilfe des Programms `makeindex` erzeugt werden kann. Daher wird von fomdss während der Dokumentkompilierung die Meldung

```
**************************************************
* PLEASE NOTE! The usage of fomdss' abbreviations
* requires the nomencl package! Please ensure that
* nomencl package is installed on your LaTeX system
* in version 4.2, dated at 2005/09/22 (or higher).
* Furthermore you have to run the program 'makeindex'
* with options:
* makeindex <file>.nlo -s nomencl.ist -o <file>.nls
**************************************************
```

ausgegeben, die zum einen den Benutzer an den Sonderstatus des Abkürzungsverzeichnisses erinnern soll und zum anderen auch direkt die weiteren Schritte zur Verzeichniserstellung aufzeigt.

abbrevrefpage

Ebenfalls im Zusammenhang mit dem Abkürzungsverzeichnis steht die Option `abbrevrefpage`. Sie kann verwendet werden, um die Seitenreferenz des ersten Auftretens der jeweiligen Abkürzung mit in das Abkürzungsverzeichnis aufzunehmen. Standardmäßig wird keine Referenzseite im Abkürzungsverzeichnis angezeigt.

Sicherlich ist jedem bewusst, dass die Option `abbrevrefpage` nur dann wirksam ist, wenn fomdss ferner mit der Option `abbreviation` geladen wird. Anderenfalls verliert diese Option ihre Bedeutung.

nopagebreak

Die Option `nopagebreak` verhindert den erzwungenen Seitenumbruch nach dem Ende jedes Verzeichnisses. Diese Option ist dann besonders nützlich, wenn am Anfang des Dokuments nur sehr kurze Verzeichnisse vorhanden sind.

Grundsätzlich sollte aber in wissenschaftlichen Arbeiten auf die Verwendung der Option `nopagebreak` verzichtet werden, da dies für diese Art Schriftstücke eher unüblich ist.

Kommandos des Dokumentbeginns

Ungeachtet der Ladeoptionen werden durch fomdss unterschiedliche Makros zur Erstellung der verschiedenen Verzeichnisse angeboten. Diese Makros sollen im weiteren Verlauf eingehender betrachtet werden.

\PrintContents

Das Makro `\PrintContents` dient der Erzeugung des Inhaltsverzeichnisses. Es verfügt nur über den optionalen Parameter `[`⟨*Überschrift*⟩`]` zur Anpassung des Verzeichnistitels. Der Standardwert ist mit *Inhalt* initialisiert. Weitere Einstellungen zur Generierung des Inhaltsverzeichnisses sind nicht notwendig!

\PrintFigures

Das Makro `\PrintFigures` dient der Erzeugung des Abbildungsverzeichnisses. Es verwendet ebenfalls nur den optionalen Parameter `[`⟨*Überschrift*⟩`]` zur Anpassung des Titels dieses Verzeichnisses. Als Standardwert wird *Abbildungen* verwendet. Auch hier sind keine weitere Einstellungen notwendig!

\PrintTables

Zur Erstellung des Tabellenverzeichnisses wird das Makro `\PrintTables` verwendet. Auch dieses Kommando unterstützt nur den einzelnen optionalen Parameter `[`⟨*Überschrift*⟩`]`, der zur Anpassung der entsprechenden Überschrift dient. Standardwert hierfür ist *Tabellen*. Zusätzliche Einstellungen können nicht vorgenommen werden!

Die Darstellung von Beispielen für die soeben beschriebenen Makros ist an dieser Stelle leider nur schwerlich möglich. Und dennoch sind innerhalb dieser Arbeit entsprechende Anschauungsobjekte vorhanden. Denn alle korrespondierenden Verzeichnisse dieses Dokuments wurden mit Hilfe der Makros `\PrintContents`, `\PrintFigures` und `\PrintTables` erstellt!

Wurde hingegen fomdss zuvor mit der Option `abbreviation` geladen, dann stehen dem Anwender spezielle Makros zur Erzeugung des Abkürzungsverzeichnisses zur Verfügung.

\PrintAbbrev

Möchte der Anwender auch ein Abkürzungsverzeichnis erzeugen, dann kann er hierfür das Makro `\PrintAbbrev` verwenden. Auch dieser Makro verfügt über den optionalen Parameter `[`⟨*Überschrift*⟩`]` zur Anpassung des Titels. Als Standardtitel wird *Abkürzungen* verwendet.

Im Gegensatz zur Erstellung der Verzeichnisse für Inhalt, Abbildungen und Tabellen muss der Benutzer die Inhalte des Abkürzungsverzeichnisses eigenständig eintragen. Dabei besteht seine Aufgabe darin, die im laufenden Text benutzten Abkürzungen bei ihrer ersten Verwendung einmalig zu definieren. Zu diesem Zweck verwendet er dann die im Folgenden gezeigten Kommandos.

\Abbrev

Mit Hilfe des Makros `\Abbrev{`⟨*Abkürzung*⟩`}{`⟨*Beschreibung*⟩`}` werden innerhalb des normalen Textes die Passagen markiert, die später in das Abkürzungsverzeichnis übernommen werden sollen. Dabei beinhaltet der erste Parameter die Abkürzung selbst und der zweite Parameter enthält die Bedeutung der im ersten Parameter angegebenen Abkürzung. Siehe hierzu nachfolgendes Beispiel.

\Mark

Das Makro `\Mark{`⟨*Buchstabe*⟩`}` dient der Hervorhebung der relevanten Buchstaben innerhalb der Abkürzungsbeschreibung. Alle hierdurch eingefassten Buchstaben werden später im Abkürzungsverzeichnis unterstrichen dargestellt.

Das Zusammenspiel der Makros `\PrintAbbrev`, `\Abbrev` und `\Mark` lässt sich am besten anhand einer beispielhaften Anwendung zeigen. Daher sollen nun insgesamt zwei Beispiele für etwas mehr Aufhellung im Umgang mit dem Abkürzungsverzeichnis sorgen.

Beispiel zur Erstellung eines Abkürzungsverzeichnisses

Das Beispiel in Quelltext 7 illustriert, wie ein Anwender die Makros `\Abbrev` und `\Mark` verwenden soll, um mit Hilfe des Makros `\PrintAbbrev` hieraus ein aus zwei Einträgen bestehendes Abkürzungsverzeichnis zu erzeugen, dessen Überschrift *Verwendete Abkürzungen* lautet.

Quelltext 7: Erzeugung des Abkürzungsverzeichnisses

```
\documentclass{fomsdt}
\usepackage[abbreviation]{fomdss}
\begin{document}
% Erzeugung des Abkürzungsverzeichnisses
\PrintAbbrev[Verwendete Abkürzungen]
% Verwendung im Textteil
... Der \Abbrev{IC}{\Mark{I}ntegrated \Mark{C}ircuit} ist
einer der Kernkomponenten moderner Computersystem ...
... Die \Abbrev{PIN}{\Mark{P}ersonal \Mark{I}dentification
\Mark{N}umber} wird zur Authentifizierung verwendet ...
\end{document}
```

Nennenswert ist hierbei, dass die vollständige Definition eines für das Abkürzungsverzeichnis bestimmten Eintrages im laufenden Text vollzogen wird. Dies verbessert nicht unbedingt die Lesbarkeit des Dokumentenquelltextes, gehört aber zu den in LaTeX üblichen Methoden. Das Makro `\footnote` sei hierfür als Beispiel genannt.

Nach Erzeugung des durch Quelltext 7 beschriebenen Dokuments enthält das Abkürzungsverzeichnis den in Abbildung 14 dargestellten Inhalt.

Verwendete Abkürzungen

IC .... Integrated Circuit

PIN .... Personal Identification Number

Abbildung 14: Ausgabe des Abkürzungsverzeichnisses

Beispiel zur Anpassung des Abkürzungsverzeichnisses

Die Anpassung des Layouts des Abkürzungsverzeichnisses ist ebenfalls gegeben. Hierzu können einfach die vom Paket nomencl angebotenen Makros verwendet werden. Einige nützliche und hilfreiche Beispiele zur Anwendung der Kommandos aus nomencl sind auch Teil der Benutzerdokumentation des Pakets fomsdt.

Ist es beispielsweise gewünscht, die Punktfüllung zwischen der Abkürzung und ihrer Erläuterung zu entfernen und soll dann auch die Abkürzung selbst als schräg gestellter Text dargestellt werden, dann kann dies auf einfache Weise durch die Manipulation des Kommandos `\nomlabel` erreicht werden. Wenn zusätzlich noch „fett“ statt „unterstrichen“ zur Hervorhebung der einzeln markierten Erläuterungsbuchstaben verwendet werden soll, dann wird hierfür einfach das Kommando `\Mark` neu definiert.

All diese wortreich beschriebenen Anpassungen können durch schlichtes Einfügen zweier Zeilen Code in das aus Quelltext 7 bekannte Beispiel erreicht werden. Das daraus entstandene Ergebnis zeigt der Codeschnipsel in Quelltext 8.

Quelltext 8: Layoutanpassungen im Abkürzungsverzeichnis

```
\documentclass{fomsdt}
\usepackage[abbreviation]{fomdss}
\begin{document}
% Abkürzungslabel ändern (schräg und ohne Punkte}
\renewcommand{\nomlabel}[1]{\textsl{#1}}
% Markierung der Abkürzung in 'fett' ändern
\renewcommand{\Mark}[1]{\textbf{#1}}
% Erzeugung des Abkürzungsverzeichnisses
\PrintAbbrev[Verwendete Abkürzungen]
% Verwendung im Textteil
... Der \Abbrev{IC}{\Mark{I}ntegrated \Mark{C}ircuit} ist
einer der Kernkomponenten moderner Computersystem ...
... Die \Abbrev{PIN}{\Mark{P}ersonal \Mark{I}dentification
\Mark{N}umber} wird zur Authentifizierung verwendet ...
\end{document}
```

Nach erneuter Kompilierung des durch Quelltext 8 beschriebenen Dokuments erhält das Abkürzungsverzeichnis das in Abbildung 15 gezeigte neue Layout.

Verwendete Abkürzungen

IC	**I**ntegrated **C**ircuit
PIN	**P**ersonal **I**dentification **N**umber

Abbildung 15: Abkürzungsverzeichnis mit geändertem Layout

Abschließend sollte noch darauf hingewiesen werden, dass die gewählte Methode zur Erzeugung des Abkürzungsverzeichnisses nicht unbedingt zu den elegantesten Varianten zählt, da zum Beispiel auch eine Umsetzung auf Basis einer Art Bibliothek hätte erfolgen können.

Dennoch spricht die relativ einfache Anwendung der Makros `\Abbrev`, `\Mark` und `\PrintAbbrev` für die gewählte und auf dem Paket nomencl basierende Implementierungsvariante, da der Endanwender hierdurch nicht genötigt wird, innerhalb seiner Abhandlung eine weitere Datei pflegen zu müssen.

3.2.3 Dokumentenende

Das Ende einer wissenschaftlichen Abhandlung verfügt, wie auch der Abhandlungsbeginn, über typische Verzeichnisse mit nützlichen Informationen für den

Leser. Die Bereitstellung der Makros zur Erzeugung und Verwaltung dieser Verzeichnisse ist Aufgabe der Datei `fomdes.sty`. Dabei bedeuten die Buchstaben des Dateinamens im Einzelnen *FOM Document End Section*.

Optionen des Dokumentenendes

Zur Konfiguration und Anpassung des Aussehens der Verzeichnisse des Dokumentenendes stehen dem Anwender auch hier einige nützliche Optionen zur Verfügung, die nun der Reihe nach vorgestellt werden sollen.

`listings`

Die Option `listings` aktiviert die Verwendung des Verzeichnisses für Codelistings, da dieses Verzeichnis per Grundeinstellung deaktiviert ist. Notwendig ist diese Option deshalb, da anderenfalls die Möglichkeit der Erzeugung eines leeren Verzeichnisses bestünde, wenn die Abhandlung keine entsprechenden Quellcodes enthält. Darum muss die Verwendung des Listingverzeichnis vom Anwender explizit gewollt sein.

`glossary`

Die Option `glossary` aktiviert die Möglichkeit zur Erstellung eines Glossars, das häufig auch als Sachwortverzeichnis bezeichnet wird. Als Grundeinstellung ist aber die Verwendung des Glossars ausgeschaltet, da hierdurch eine Abhängigkeit zum Paket gloss aufgebaut wird. Leider gehört auch das Paket gloss nicht zum Standardumfang der verschiedenen LaTeX-Distributionen und ist somit eventuell noch nachträglich zu installieren.

Falls nun ein Anwender das Paket gloss nicht auf seinem System installiert hat, dann darf die Abhängigkeit zum Paket gloss keine Fehlermeldungen hervorbringen. Dies ist auch der Grund, warum der Anwender die Verwendung des Glossars ausdrücklich aktivieren muss. Im Falle der Aktivierung wird von fomdes während der Kompilierung eines Dokuments die Meldung

```
*************************************************
* PLEASE NOTE! The usage of a glossary requires to
* run: bibtex <file>.gls
*************************************************
```

ausgegeben. Diese Paketmeldung soll den Anwender darauf hinweisen, dass zur endgültigen Erzeugung des Glossars auch ein Aufruf des Kommandos `bibtex` erforderlich wird.

`glossrefpage`

Ebenfalls im Zusammenhang mit der Glossarerstellung steht die Option `glossrefpage`. Sie wird dazu eingesetzt, auch die Seitenzahl eines Glossarbegriffs anzuzeigen. Standardmäßig wird keine Referenzseite im Glossar verwendet.

Sicherlich ist jedem bewusst, dass die Option `glossrefpage` nur dann wirksam ist, wenn fomdes ferner mit der Option `glossary` geladen wird. Anderenfalls verliert diese Option ihre Bedeutung.

nobibliography

Das Literaturverzeichnis ist ein Muss einer jeden wissenschaftlichen Abhandlung und sollte daher in diesen Arbeiten niemals ausgelassen werden. Gibt es dennoch einmal einen Grund, dass die Verwendung des Literaturverzeichnisses unterdrückt werden muss, dann kann zu diesem Zweck die Option `nobibliography` verwendet werden.

Wird hingegen diese Option *nicht* in den Stilladeparametern angegeben, dann wird durch fomdes die Meldung

```
**************************************************
* PLEASE NOTE! The usage of a bibliography requires
* to run: bibtex <file>.aux
**************************************************
```

ausgegeben. Diese Meldung dient dem Anwender als Hinweis, dass zur endgültigen Erstellung des Literaturverzeichnisses weitere Eingriffe erforderlich sind.

index

Auch die Erstellung des Indexes beziehungsweise des Stichwortverzeichnisses muss durch den Anwender zuvor aktiviert werden, da per Standardeinstellung die Indexerzeugung ausgeschaltet ist. Die Option `index` ermöglicht dem Endanwender diese Aktivierung.

Wurde in den Ladeparametern von fomdes die Option `index` angegeben, dann wird während der Dokumenterstellung der Benutzer durch die Meldung

```
**************************************************
* PLEASE NOTE! Index creation requires to run:
* makeindex <file>.idx
* But it is recommended to create an index based on
* ISO style. For this run:
* makeindex <file>.idx -s fomidx.ist
* Especially for German index creation users should
* use option '-g' to ensure German word ordering
* rules according to DIN 5007. In this case run:
* makeindex <file>.idx -g -s fomidx.ist
**************************************************
```

daran erinnert, dass noch der Aufruf des Programms `makeindex` erforderlich ist, um die tatsächliche Generierung des Indexes durchzuführen.

nopagebreak
Die Option `nopagebreak` verhindert den erzwungenen Seitenumbruch nach dem Ende jedes Verzeichnisses. Diese Option ist dann besonders nützlich, wenn sich am Dokumentenende nur sehr kurze Verzeichnisse finden lassen.

Grundsätzlich sollte in wissenschaftlichen Arbeiten jedoch auf die Anwendung der Option `nopagebreak` verzichtet werden, da dies in der Regel die Übersichtlichkeit stört und dem Leser keinen nennenswerten Vorteil bietet.

Kommandos des Dokumentenendes
In Abhängigkeit der angegebenen Ladeoptionen stehen dem Benutzer verschiedene Makros zur Erzeugung der jeweiligen Verzeichnisse zur Verfügung. Die Erläuterung der einzelnen Makros soll Gegenstand des hier anschließenden Teilabschnittes sein. Vorab jedoch noch einige Worte zum allgemeinen Verhalten aller hier vorgestellten Verzeichnismakros.

Wird einer oder auch mehrere der Parameter `listings`, `glossary`, `nobibliography` und `index` in den Ladeoptionen von fomdes nicht angegeben, dann bleiben dennoch alle korrespondierenden Makros erhalten. In diesem Fall enthalten die betroffenen Makros keinerlei Funktionalität!

Dieses Verhalten wurde implementiert, damit der Anwender nur an einer Stelle seiner Dokumentquellen Änderungen vornehmen muss, wenn er gedenkt, eines der Verzeichnisse auszublenden. Das erleichtert den praktischen Umgang mit fomdes enorm, da es hierdurch nicht mehr zu der oft lästigen Fehlermeldung „Undefined control sequence“ kommt, wenn der Endanwender vergaß, das zugehörige Makro ebenfalls auszuschalten.

\PrintListings
Das Makro `\PrintListings` dient der Erzeugung des Verzeichnisses für Codelistings. Es verfügt über den optionalen Parameter [⟨*Überschrift*⟩], der zur Änderung des Standardtitels *Listings* verwendet wird. Weitere Einstellungen zur Erzeugung des Listingverzeichnisses sind nicht notwendig! Jedoch erfolgt eine tatsächliche Verzeichniserstellung erst durch die Angabe des Parameters `listings` in den fomdes-Ladeoptionen. Anderenfalls bleibt die Verwendung von `\PrintListings` wirkungslos.

\PrintGlossary
Das Makro `\PrintGlossary` wird zum Zweck der Erstellung des Glossars verwendet. Es verfügt im Gegensatz zum vorherigen Makro über insgesamt zwei optionale Argumente sowie über einen Pflichtparameter.

Analog zum vorherigen Makro dient der erste optionale Parameter [⟨*Überschrift*⟩] der Anpassung des Glossartitels, dessen Standardwert mit *Glossar* initialisiert wird.

Weiterhin besteht die Möglichkeit der Übernahme aller in der Glossarbibliothek befindlichen Einträge in das Ausgabedokument. Hierfür kann als zweiter optionaler Parameter der Wert `[all]` angegeben werden. Nur ist zu beachten, dass das Paket gloss im Falle der Verwendung aller Glossareinträge die Warnmeldung

```
Package gloss Warning: Term '*' on page ⟨n⟩ not
defined in glossary 'default' on input line ⟨m⟩
```

ausgibt! Daher ist es nicht unbedingt empfehlenswert, alle in der Glossarbibliothek verfügbaren Einträge auch in das Ausgabedokument zu übertragen.

Der dritte und letzte Parameter des Makros `\PrintGlossary` ist als Pflichtargument implementiert und dient der Bekanntgabe der zu verwendenden Bibliotheksdatei. Dieser Dateiname muss ohne Dateierweiterung angegeben werden, da anderenfalls die entsprechende Bibliothek nicht gefunden werden kann.

Abschließend gilt es noch zum Makro `\PrintGlossary` anzumerken, dass die Umsetzung der automatisierten Glossarerzeugung auch ohne eine Abhängigkeit zum Paket gloss hätte erfolgen können. Dennoch sprachen einige Gründe für den eingeschlagenen Weg!

Zum einen verfügte bereits die Standardkonfiguration der durch Paket gloss bereitgestellten Glossareinträge über ein sehr ansprechendes Layout, das nur in wenigen Nuancen angepasst werden musste, und zum andern zählt die bibliotheksbasierte Erstellung des Sachwortverzeichnisses zu den mit Abstand elegantesten Varianten möglicher Verfahren. Daher wurde dieser Teil der Implementierung auf der Grundlage des Pakets gloss durchgeführt, auch auf die Gefahr hin, dass für eine fehlerfreie Glossarerstellung eventuell die nachträgliche Paketinstallation durch den Anwender unumgänglich ist.

\PrintBibliography

Das Makro `\PrintBibliography` dient der Erzeugung des so wichtigen Literaturverzeichnisses. Es verwendet ebenfalls insgesamt drei Parameter, von denen die ersten beiden als freiwillige Argumente und das dritte als Pflichtparameter ausgelegt sind.

Auch hier kann der Titel des Verzeichnisses mit Hilfe des ersten optionalen Parameters [⟨*Überschrift*⟩] eingestellt werden, dessen Standardwert auf *Literatur* festgelegt ist.

Hingegen kann der zweite optionale Parameter zur Optimierung des Layouts des Literaturverzeichnisses verwendet werden. Der hier angegebene Wert wird dann von fomdes ohne weitere Prüfung an das LaTeX-eigene Makro `\bibliographystyle` übergeben. Somit sollte der Anwender zumindest ein gewisses Grundverständnis dafür besitzen, was hier anzugeben ist. Der Standardwert

dieses optionalen Parameters ist *dinat*, also der Dateiname `dinat.bst` ohne Dateiendung. Dennoch wird von der Verwendung dieses Parameters abgeraten, da durch die Datei `dinat.bst` bereits alle notwendigen Einstellungen für ein DIN-gerechtes Literaturverzeichnis vorgenommen werden!

Die Bekanntgabe der Bibliotheksdatei ist auch hier dem dritten Makroparameter, also dem Pflichtargument, vorbehalten. Dieser Dateiname muss ebenfalls ohne Dateierweiterung angegeben werden, da ansonsten die entsprechende Bibliothek vom LaTeX-System nicht gefunden wird.

\PrintIndex

Das Makro `\PrintIndex` ist das letzte Kommando, das durch fomdes zur Verzeichniserstellung angeboten wird. Es verfügt nur über insgesamt zwei optionale Parameter, die einer Anpassung des Verzeichnislayouts dienen.

Wie bei allen zuvor aufgeführten Makros kann auch hier der erste optionale Parameter [⟨*Überschrift*⟩] zur Änderung des Verzeichnistitels verwendet werden. Die Standardüberschrift des Stichwortverzeichnisses ist *Index*.

Mit Hilfe des zweiten freiwilligen Arguments [⟨*Spalten*⟩] ist eine Variation der Spaltenzahl des Indexes gegeben. Dabei können hier die gültigen Werte wie folgt angegeben werden:

`single` Wird `\PrintIndex`[⟨*Überschrift*⟩][single] verwendet, dann besteht der Index aus nur einer Spalte.

`double` Wird `\PrintIndex`[⟨*Überschrift*⟩][double] verwendet, dann wird der Index auf zwei Spalten verteilt.

`triple` Wird `\PrintIndex`[⟨*Überschrift*⟩][triple] verwendet, dann stehen dem Index insgesamt drei Spalten zur Verfügung.

Wird der optionale Parameter [⟨*Spalten*⟩] nicht angegeben, dann verwendet der Index *double* als Standardwert für die Anzahl der Spalten und die Angabe eines fehlerhaften Arguments führt zur Augabe der Warnmeldung

```
Package fomdes Warning: PrintIndex: Unknown
column option '⟨value⟩'. Use 2 columns instead.
```

Leider ist auch an dieser Stelle die beispielhafte Demonstration der Verwendung der soeben beschriebenen Makros nur schwerlich möglich. Jedoch sei angemerkt, dass die Erstellung der am Ende dieser Abhandlung befindlichen Verzeichnisse mit Hilfe der zuvor vorgestellten Makros `\PrintListings`, `\PrintGlossary`, `\PrintBibliography` und `\PrintIndex` erfolgte.

`\SeeAlso`

Das Makro `\SeeAlso{`⟨*Querverweis*⟩`}` ist für die einheitliche Formatierung von Verweisen gedacht und innerhalb eines Indexes und eines Glossars ebenso nützlich wie hilfreich, da es einen kleinen Pfeil mit dahinter befindlichen schräg gestelltem Text erzeugt. Diese Art der Formatierung soll der besseren Hervorhebung der entsprechenden Verweise dienen.

Mit Hilfe des Parameters `{`⟨*Querverweis*⟩`}` kann nicht nur reiner Text verarbeitet werden! Vielmehr besteht auch die Möglichkeit, weitere Makros einzubinden, wie das kleine Beispiel in Quelltext 9 demonstriert.

Quelltext 9: Verwendung des Makros `\SeeAlso`

```
... Dieser Monitor\index{Monitor\SeeAlso{\textbf{Bildschirm}}}
zeigt die Zustandsparameter des Gesamtsystems, ...
```

Nach erneuter Kompilierung des auf Quelltext 9 basierenden Dokuments stellt sich das fertig formatierte Ausgabeergebnis des Stichwortverzeichnisses wie in Abbildung 16 gezeigt dar.

M

Monitor → ***Bildschirm*** . . . . . . ⟨*Seite*⟩
. . .

Abbildung 16: Ergebnis der Makroverwendung `\SeeAlso`

Praktisch bestünde auch die Möglichkeit, das Makro `\SeeAlso` im laufenden Text zu verwenden. Jedoch wäre dies wenig sinnvoll, wenn hierfür keine entsprechende Referenz verfügbar ist.

3.2.4 Absatzdefinition

Die spezielle Formatierung von Textpassagen ist mit LaTeX nicht gerade einfach zu realisieren. Um nun dem Endanwender, und hier speziell den LaTeX-Neulingen, eine einfache Formatanpassung von Fließtexten zu ermöglichen, wurde dem Paket fomsdt die Datei `fompar.sty` beigefügt. Dabei bedeuten die Buchstaben des Dateinamens im Einzelnen *FOM Paragraph Definition*.

Die Angabe von Dateiladeoptionen ist hier nicht erforderlich, da alle bereitgestellten Makros so definiert sind, dass sie unabhängig von etwaigen Randbedingungen angewendet werden können.

Kommandos zur Absatzformatierung

Die in der Datei `fompar.sty` implementierten Kommandos, dazu zählen drei Kernmakros und eine Umgebung, orientieren sich allesamt an grundlegenden Belangen zur textlichen Formatierung innerhalb von wissenschaftlichen Abhandlungen. Sie sind daher sehr stark an die Bedürfnisse angepasst, die im Zusammenhang mit einer praktischen Erstellung derartiger Arbeiten auftreten.

Citation

In wissenschaftlichen Arbeiten werden in der Regel längere Zitate in eigenen und zusammenhängenden Absätzen dargestellt. Dabei sind diese Zitate nicht nur durch doppelte Anführungszeichen hervorgehoben, sondern verfügen oft auch über einen zusätzlichen linken und rechten Absatzeinzug. Die Formatierung dieser Art Textpassagen ist Aufgabe der Umgebung `Citation`.

Neben der optionalen Angabe des Abstandes zum umgebenden Text, die Standardeinstellung hierfür ist der Wert von *parskip*, muss zur Verwendung dieser Umgebung auch der linke und rechte Einzug angegeben werden, wie das Beispiel in Quelltext 10 zeigt.

Quelltext 10: Verwendung der Umgebung `Citation`

```
... vorhergehender Text.
\begin{Citation} % Beginn der Umgebung
      {1cm}      % Linker Einzug
      {3cm}      % Rechter Einzug
"‘Text des Zitats ..."’
\end{Citation}   % Ende der Umgebung
Nachfolgender Text ...
```

Nach Erzeugung des Dokuments, welches das in Quelltext 10 gezeigte Codefragment enthält, ergibt sich das in Abbildung 17 dargestellte Ausgabeergebnis.

> ... vorhergehender Text.
>
> „Text des Zitats ...“
>
> Nachfolgender Text ...

Abbildung 17: Ergebnis der Verwendung der Umgebung `Citation`

Leider ist der unterschiedlich große Einzug auf beiden Seiten des Zitatabsatzes in Abbildung 17 nur schwerlich zu erkennen. Daher der Hinweis, dass die Angabe des linken und rechten Absatzeinzuges unabhängig voneinander erfolgt und beide Werte auch getrennt voneinander verarbeitet werden. In der praktischen Anwendung enthalten beide Parameter jedoch zumeist identische Werte.

`\Remark` und `\Item`

Die Makros `\Remark` und `\Item` dienen beide der Formatierung von Absätzen, die über einen links stehenden Bezeichner und einen davon rechts befindlichen Textkörper verfügen. Da sich beide Makros im Grunde nur in der Ausrichtung ihres Bezeichners unterscheiden, erscheint deren gemeinsame Behandlung als sinnvoll. Schließlich soll der geneigt Leser nicht dazu gezwungen werden, identischen Inhalt doppelt lesen zu müssen.

Beide Makros verfügen über insgesamt vier Parameter, von denen die ersten beiden optionale und die letzten beiden obligatorische Informationen enthalten.

Als erstes freiwilliges Argument kann der [⟨*Einzug*⟩] vom linken Rand an die Makros `\Remark` und `\Item` übergeben werden. Somit steuert dieser Wert den linksseitigen Abstand des Textkörpers zum Seitenrand. Wird dieses Argument nicht angegeben, dann wird für `\Remark` ein Standardeinzug von *3cm* gewählt und bei `\Item` der Standardeinzug auf *4cm* voreingestellt.

Der zweite freiwillige Parameter dient den Makros `\Remark` und `\Item` als Definition des [⟨*Abstandes*⟩] zum umgebenden Text. Somit kann auch die vertikale Ausrichtung des Textkörpers zum restlichen Text beeinflusst werden. Wird dieser Abstand nicht gesetzt, dann benutzen beide Makros den aktuellen Wert von *parskip* als Standardeinstellung.

Beide Pflichtargumente werden dazu verwendet, den eigentlichen Inhalt der jeweiligen Absätze anzugeben. Dabei sollte aber beachtet werden, dass der Text des ersten Parameters {⟨*Bezeichner*⟩} nicht auf mehrere Zeilen verteilt wird, wenn dieser über die Breite des verwendeten Einzugs hinaus geht.

Hingegen wird der Textkörper selbst wie gewohnt am Zeilenende umbrochen. Es ist sogar möglich, mehrere Unterabsätze im Textkörper anzugeben, die dann gleichfalls um den im Einzug angegebenen Wert eingerückt werden.

Weiterhin ist es möglich und sehr häufig auch empfehlenswert, den Bezeichner eines durch die Makros `\Remark` und `\Item` formatierten Absatzes noch zusätzlich hervorzuheben. Daher wurde während der Implementierung darauf geachtet, dass diese Textmanipulationen gleichfalls in der Ausgabe erscheinen.

Beispiele zur Verwendung der Makros `\Remark` und `\Item`

Das erste Beispiel soll zeigen, wie das Makro `\Remark` verwendet werden kann, um beispielsweise eine besonders wichtige Textpassage in einer wissenschaftlichen Arbeit kenntlich zu machen. Zu diesem Zweck wird angenommen, dass der Abhandlungsautor den Leser darauf hinweisen möchte, welche unter Umständen negativen Folgen die Anwendung eines bestimmten Befehls haben kann. Quelltext 11 zeigt den zur Formatierung dieses Hinweises nötigen Code.

Quelltext 11: Verwendung des Makros `\Remark`

```
\Remark[25mm]{\bf Bemerkung:}{Wird das Kommando {\tt dd}
       zusammen mit den Optionen {\tt if=/dev/hda},
       {\tt of=/mnt/floppy/mbr}, {\tt bs=512} und
       {\tt count=1} verwendet, dann kann der
       Bootsektor einer Festplatte auf Diskette
       gespeichert werden.}
\Remark[25mm]{\bf Achtung!}{Von der Verwendung des Befehls
       {\tt dd} mit den Parametern {\tt if=/dev/zero}
       und {\tt of=/dev/hda} wird dringend abgeraten!}
```

Nach erfolgter Kompilierung des Dokuments, welches den Code aus Quelltext 11 enthält, stellt sich das Ausgabeergebnis wie in Abbildung 18 gezeigt dar.

Bemerkung:	Wird das Kommando `dd` zusammen mit den Optionen `if=/dev/hda`, `of=/mnt/floppy/mbr`, `bs=512` und `count=1` verwendet, dann kann der Bootsektor einer Festplatte auf Diskette gespeichert werden.
Achtung!	Von der Verwendung des Befehls `dd` mit den Parametern `if=/dev/zero` und `of=/dev/hda` wird dringend abgeraten!

Abbildung 18: Ergebnis der Makroverwendung `\Remark`

Das zweite Beispiel verdeutlicht, wie mit Hilfe des Makros `\Item` eine listenartige Strukturierung aufgebaut werden kann. Dabei basiert das in Quelltext 12 verwendete Beispiel auf der Annahme, dass der Abhandlungsautor einige seiner eigenen Kommandos näher erläutern möchte.

Quelltext 12: Verwendung des Makros `\Item`

```
\Item{\tt create}{Die Funktion {\tt create} reserviert alle
     Systemressourcen und weist den benötigten Speicher zu.}
\Item{\tt connect}{Die Funktion {\tt connect} stellt eine
     Verbindung zum Server her.}
\Item{\tt disconnect}{Die Funktion {\tt disconnect} baut die
     zuvor hergestellte Verbindung zum Server wieder ab.}
\Item{\tt destroy}{Die Funktion {\tt destroy} gibt alle
     reservierten Systemressourcen und den zugewiesenen
     Speicher wieder frei.}
```

Die Erstellung des Dokuments, das den in Quelltext 12 gezeigten Code enthält, ermöglicht das in Abbildung 19 darstellte Ausgabeergebnis.

Wie bereits eingangs erwähnt und durch die Abbildungen 18 und 19 belegt, unterscheiden sich die Makros `\Remark` und `\Item` letztlich nur in der Ausrichtung ihres Bezeichners. Dieser Umstand wirft nun unweigerlich die Frage auf, warum

`create`	Die Funktion `create` reserviert alle Systemressourcen und weist den benötigten Speicher zu.
`connect`	Die Funktion `connect` stellt eine Verbindung zum Server her.
`disconnect`	Die Funktion `disconnect` baut die zuvor hergestellte Verbindung zum Server wieder ab.
`destroy`	Die Funktion `destroy` gibt alle reservierten Systemressourcen und den zugewiesenen Speicher wieder frei.

Abbildung 19: Ergebnis der Makroverwendung `\Item`

nicht nur eines der beiden Makros entwickelt wurde. Die Antwort darauf findet sich in erster Linie in der Verwendung der optionalen Makroparameter.

Würde als Beispiel ein zusätzlicher freiwilliger Parameter zur Ausrichtung des Bezeichners vorhanden sein, dann entstünde hieraus eine unnötige Komplexität! Da in LaTeX zwar die Implementierung optionaler Parameter möglich ist, ist dennoch ihre Reihenfolge durch das System festgelegt. Das bedeutet nichts anderes, als dass bereits zur Verwendung des zweiten optionalen Parameters die Angabe des ersten optionalen Parameters erforderlich wird. Daraus folgt, dass die Abwägung, welches der freiwilligen Argumente nun wichtiger als ein anderes ist und somit in der Reihenfolge höher priorisiert werden kann, für den hier diskutierten Anwendungsfall nur schwer möglich ist. Daher wurde während der Umsetzung entschieden, auch hier das sinnvolle Limit von maximal zwei optionalen Parametern nicht zu überschreiten.

Konfiguration der Makros `\Remark` und `\Item`

Während der Implementierung der Datei `fompar.sty` konnte festgestellt werden, dass eine global wirkende Konfiguration des Einzugs sowie des Abstandes zum umgebenden Text sinnvoll sein könnte. Zu diesem Zweck wurden weitere Makros implementiert. Da diese Makros ihre Daten analog verarbeiten, soll an dieser Stelle nur kurz darauf eingegangen werden.

Die Makros `\RemarkIndent` und `\ItemIndent` dienen der globalen Anpassung des Absatzeinzuges der korrespondierenden Makros `\Remark` und `\Item` und sind zur alternativen Verwendung zum ersten optionalen Parameter gedacht.

Mit Hilfe der Makros `\DefaultRemarkIndent` und `\DefaultItemIndent` kann ein zuvor global geänderter Absatzeinzug auf seinen Standardwert zurückgesetzt werden.

Ähnlich verhält es sich mit den Makros `\RemarkOffset` und `\ItemOffset`, die als Alternative zum zweiten optionalen Parameter gedacht sind, und den Makros `\DefaultRemarkOffset` und `\DefaultItemOffset`, welche das Zurücksetzen der jeweiligen Standardwerte erlauben. Nur werden diese Makros zur Anpassung des Abstandes zum umgebenden Text verwendet.

Zur Demonstration der Anwendung der hier vorgestellten Makros soll nun das Beispiel aus Quelltext 11 herangezogen und entsprechend modifiziert werden.

Quelltext 13: Globale Anpassung des Einzugs für Makro `\Remark`

```
\RemarkIndent{55mm}
\Remark{\bf Bemerkung:}{Wird das Kommando {\tt dd}
       zusammen mit den Optionen {\tt if=/dev/hda},
       {\tt of=/mnt/floppy/mbr}, {\tt bs=512} und
       {\tt count=1} verwendet, dann kann der
       Bootsektor einer Festplatte auf Diskette
       gespeichert werden.}
\Remark{\bf Achtung!}{Von der Verwendung des Befehls
       {\tt dd} mit den Parametern {\tt if=/dev/zero}
       und {\tt of=/dev/hda} wird dringend abgeraten!}
\DefaultRemarkIndent
```

Nach der Erzeugung des Dokuments, welches den Code aus Quelltext 13 enthält, stellt sich das Ausgabeergebnis wie in Abbildung 19 gezeigt dar.

Bemerkung:	Wird das Kommando `dd` zusammen mit den Optionen `if=/dev/hda`, `of=/mnt/floppy/mbr`, `bs=512` und `count=1` verwendet, dann kann der Bootsektor einer Festplatte auf Diskette gespeichert werden.
Achtung!	Von der Verwendung des Befehls `dd` mit den Parametern `if=/dev/zero` und `of=/dev/hda` wird dringend abgeraten!

Abbildung 20: Ergebnis der globalen Anpassung für Makro `\Remark`

Erst durch diese Konfigurationsmakros wird es möglich, sowohl den Einzug als auch den Abstand der Makros `\Remark` und `\Item` einheitlich zu setzen, ohne dafür jedes einzelne optionale Argument ändern zu müssen!

`\Description`

Die allgemeingültige Reglementierung, dass die Gliederungstiefe einer wissenschaftlichen Abhandlung nicht mehr als drei Ebenen umfassen darf, verursacht

mancherorts einige Schwierigkeiten. Und dennoch kann sich eine sinnvoll gewählte Substrukturierung einzelner Kapitel positiv auf die Lesbarkeit des gesamten Werkes auswirken [21, Seite 8].

Daher bietet fompar das Makro `\Description` an, mit dessen Hilfe die Erzeugung von Pseudoüberschriften ermöglicht wird. Sicher wird sich der ein oder andere LaTeX-kundige Leser fragen, warum hierfür nicht die Makros `\paragraph` und `\subparagraph` verwendet werden.

Die Beantwortung dieser Frage ist simpel und lautet: Beide Makros gehören zu den in LaTeX verfügbaren Gliederungskommandos! Somit gehen sie einher mit den Makros `\part`, `\chapter`, `\section` und so weiter. Dass die Überschriften der Makros `\paragraph` und `\subparagraph` nicht im Inhaltsverzeichnis auftauchen, liegt einzig und allein daran, dass in LaTeX die Standardgliederungstiefe auf drei Ebenen eingestellt ist. Denn die einfache Manipulation der beiden Zähler `tocdepth` und `secnumdepth` veranlasst LaTeX, die durchnummerierten Inhalte von `\paragraph` und `\subparagraph` ebenfalls mit im Inhaltsverzeichnis aufzunehmen!

Damit die Eintragung der verschiedenen Teilabschnitte eines Kapitels im Inhaltsverzeichnis von vornherein ausgeschlossen werden kann, sollte immer das fompar-Makro `\Description` bevorzugt werden, anstatt die Makros `\paragraph` und `\subparagraph` zu verwenden.

Zur Benutzung des Makros `\Description` stehen dem Endanwender, neben einem freiwilligen Argument, auch zwei obligatorische Parameter zur Verfügung, die zur Bekanntgabe des Inhalts des entsprechenden Abschnittes dienlich sind. Dabei definiert der erste Pflichtparameter {⟨*Überschrift*⟩} den Kopf und der zweite Pflichtparameter {⟨*Text*⟩} den Körper des entsprechenden Absatzes.

Der optionale Parameter [⟨*Format*⟩] kann zur Formatierung der Absatzüberschrift verwendet werden. Hierfür stehen dann verschiedene vordefinierte Werte bereit, die hier kurz erläutert werden sollen.

`n` Die Absatzüberschrift wird nicht formatiert.

`b` Die Absatzüberschrift wird **fett** dargestellt (Standardeinstellung).

`s` Die Absatzüberschrift wird *schräg gestellt* dargestellt.

`u` Die Absatzüberschrift wird unterstrichen dargestellt.

Was noch fehlt, ist ein Beispiel. Da die Auswirkungen des Makros `\Description` auf das Absatzlayout eher weniger spektakulär sind, werden die verschiedenen Möglichkeiten in nur einem Beispiel zusammengefasst. Hierdurch sind dann auch die jeweiligen Unterschiede besser erkennbar.

Quelltext 14: Beispiele zur Verwendung des Makros `\Description`

```
\Description[n] % Keine Formatierung
            {Ohne Formatierung}
            {Unformatierte Überschriften sind zur Hervorhebung
             meist ungeeignet.}\par
\Description[b] % Fett formatiert, kann weggelassen werden
            {Überschrift fett}
            {Als fett formatierte Überschriften sind gut zur
             Hervorhebung von Hauptabschnitten geeignet.}\par
\Description[s] % Schräg formatiert
            {Schräg gestellte Überschrift}
            {Schräg gestellte Überschriften sind gut zur
             Hervorhebung von Unterabschnitten geeignet.}\par
\Description[u] % Unterstrichen formatiert
            {Unterstrichene Überschrift}
            {Unterstreichungen jeglicher Art sollten in
             Ausarbeitungen besser vermieden werden!}
```

Nach erfolgter Kompilierung des Dokuments, das den Code aus Quelltext 14 enthält, wird das in Abbildung 21 gezeigte Layout sichtbar, dass durch Verwendung des Makros `\Description` generiert wurde.

Ohne Formatierung
Unformatierte Überschriften sind zur Hervorhebung meist ungeeignet.

Überschrift fett
Als fett formatierte Überschriften sind gut zur Hervorhebung von Hauptabschnitten geeignet.

Schräg gestellte Überschrift
Schräg gestellte Überschriften sind gut zur Hervorhebung von Unterabschnitten geeignet.

Unterstrichene Überschrift
Unterstreichungen jeglicher Art sollten in Ausarbeitungen besser vermieden werden!

Abbildung 21: Mögliche Layouts des Makros `\Description`

Abbildung 21 lässt erkennen, dass der Zwischenraum von Überschrift und Textblock kleiner ist als der Abstand zum nachfolgenden Absatz. Dieses Verhalten ist keineswegs Zufall! Denn hierdurch soll dem Leser einer Abhandlung besser kenntlich gemacht werden, zu welchem Teilabschnitt die entsprechende Überschrift gehört.

3.2.5 Gleitobjektdefinition

Besonderes Augenmerk gilt in wissenschaftlichen Arbeiten der Formatierung so genannter Gleitobjekte. Dazu zählen Abbildungen, Tabellen und im weiteren Sinne auch Codelistings. Daher bietet die Gleitobjektdefinition fomfod, die in der Datei `fomfod.sty` implementiert ist, entsprechende Makros zur einfachen und einheitlichen Integration dieser Gleitobjekte in wissenschaftlichen Abhandlungen an. Wie man sich leicht vorstellen kann, bedeuten hier die einzelnen Buchstaben des Dateinamens *FOM Floating Object Definition*.

Optionen der Gleitobjektdefinition

Zur Konfiguration und Anpassung des Aussehens der einzelnen Gleitobjekte stehen einem Anwender auch hier einige nützliche Optionen zur Verfügung, die hier erst einmal der Reihe nach vorgestellt werden sollen.

`classiccaption`

Zur Betonung eines Gleitobjektes ist es allgemein üblich, dessen Beschriftung in einer serifenlosen Schrift vorzunehmen. Diese serifenlose Schrift wird somit auch durch fomfod als Standardeinstellung für Objektbeschriftungen verwendet.

Wenn einem Anwender die Standardformatierung der Beschriftung nicht zusagt, dann kann mit Hilfe der Option `classiccaption` eine Anpassung der Beschriftungen unterdrückt werden. Dies hätte jedoch möglicherweise zur Folge, dass sich der entsprechende Objekttitel weniger vom umgebenden Text abhebt, als dies bei Verwendung einer serifenlosen Schrift der Fall ist.

`listingcolored`

Die farbliche Hervorhebung bestimmter Quelltextpassagen ist nicht selten ein geeignetes und nützliches Mittel zur Unterstreichung verschiedener Zusammenhänge. Daher wird dem Anwender die Möglichkeit geboten, die in Abhandlungen einzubindenden Quelltexte koloriert darzustellen. Diese Quellcodefärbung kann somit durch Angabe der Option `listingcolored` aktiviert werden.

Kommandos der Gleitobjektdefinition

Für die häufigsten in wissenschaftlichen Abhandlungen vorkommenden Formen von Gleitobjekten definiert die Datei `fomfod.sty` entsprechende Makros. Doch stellte sich während der Umsetzung heraus, dass sich die unterschiedlichen Gleitobjektarten nur sehr ungleich abstrahieren ließen. Dieser Umstand spiegelt sich dann auch in der Anzahl der jeweiligen Makros wieder, die nun im Anschluss einzeln vorgestellt werden sollen.

\SinglePicture

Abbildungen bilden in wissenschaftlichen Arbeiten die wohl am häufigsten verwendete Form von Gleitobjekten. Zu diesem Zweck hält die Datei `fomfod.sty` das Makro `\SinglePicture` bereit. Dieses Makros erlaubt dem Endanwender die Integration einer Abbildung auf einfachste Weise, da alle hierfür relevanten Informationen als Übergabeparameter definiert sind.

Das Makro `\SinglePicture` verfügt, neben den vier Pflichtparametern {⟨*Beschriftung*⟩}, {⟨*Referenzlabel*⟩}, {⟨*Bilddateiname*⟩} und {⟨*Quellenangabe*⟩}, auch über die beiden optionalen Parameter [⟨*Breite*⟩] und `[box]`, die zur individuellen Anpassung einer eingebundenen Graphik verwendet werden können.

Dabei erlaubt der erste optionale Parameter [⟨*Breite*⟩] die Anpassung der Bildbreite in Abhängigkeit von der aktuell eingestellten Textbreite, welche durch das LaTeX-Makro `\textwidth` vorgegeben wird. Eine Angabe der Bildhöhe ist nicht notwendig, da diese proportional skaliert wird. Der Standardwert dieses optionalen Arguments ist *0.9* und somit 90 % der durch `\textwidth` beschriebenen Breite des umgebenden Textes.

Der optionale Parameter `[box]` dient der Erzeugung eines Rahmens um die eingebundene Abbildung. Da die im vorherigen Kapitel gegebenen Empfehlungen von der Darstellung eines Abbildungsrahmens abraten, ist der Standardwert dieses Parameters so eingestellt, dass keine Rahmen für Abbildungen verwenden werden.

Da die Beschriftung einer Abbildung in wissenschaftlichen Abhandlungen ein Muss ist, kann zur Definition des jeweiligen Abbildungstitels der erste Pflichtparameter {⟨*Beschriftung*⟩} verwendet werden.

Mit Hilfe des zweiten Pflichtparameters {⟨*Referenzlabel*⟩} hat der Anwender die Möglichkeit der Vereinbarung eines Bezeichners. Dieser Bezeichner kann dann dazu verwendet werden, um die hierzu zugehörende Abbildung innerhalb des restlichen Dokumentes zu referenzieren. Üblicherweise folgt ein Abbildungslabel dem Schema `fig:label`.

Essenziell für das Einladen externer Graphiken in einer Abhandlung ist der Name der benötigten Abbildungsdatei. Hierfür kann vom Anwender der dritte obligatorische Parameter {⟨*Bilddateiname*⟩} verwendet werden. Zur Steuerung des Suchpfades zu den einzelnen Abhandlungsabbildungen empfiehlt sich auch die Verwendung des LaTeX-eigenen Makros `\graphicspath`.

Wenn die eingefügte bildliche Darstellung einer Fremdquelle entspringt, dann ist es nicht nur redlich, sonder in wissenschaftliche Arbeiten auch unumgänglich, die Herkunft der Graphik entsprechend kenntlich zu machen. Zu diesem Zweck kann der vierte und letzte Pflichtparameter {⟨*Quellenangabe*⟩} verwendet werden. Dabei erfolgt eine Quellenangabe typischerweise rechtsbündig und direkt unterhalb der jeweiligen Abbildung.

Beispiel zur Verwendung des Makros `\SinglePicture`

Ein Beispiel soll helfen, die einfache Anwendung des Makros `\SinglePicture` zu demonstrieren. Dabei wird von der Annahme ausgegangen, dass der Autor einem Leser die Anordnung der Kontakte einer Chipkarte anhand einer Abbildung, die in der Datei `kontakte.pdf` enthalten ist, näher bringen möchte. Das Fragment in Quelltext 15 zeigt den hierfür notwendigen Code.

Quelltext 15: Beispiel zur Verwendung des Makros `\SinglePicture`

```
\SinglePicture[0.7]% 70% der Textbreite
              [box]% inklusive Rahmen
              {Kontakte einer Chipkarte}% Beschriftung
              {fig:ChipKontakte}%         Referenzlabel
              {kontakte.pdf}%             Bilddateiname
              {Quelle: \url{http://www.hardwareecke.de/}}% Quellenangabe
```

Nach Erstellung des den Code aus Quelltext 15 enthaltenden Dokuments stellt sich das Ausgabeergebnis wie in Abbildung 22 gezeigt dar.

Versorgungsspannung (Vcc)
Masse (GND)
Reseteingang (RST)
Programmierspannung (Vpp)
Takteingang (CLK)
Serieller Ein- und Ausgang (I/O)
Reserviert (RFU)
Reserviert (RFU)
C1 C2 C3 C4 C5 C6 C7 C8

Quelle: `http://www.hardwareecke.de/`

Abbildung 1: Kontakte einer Chipkarte

Abbildung 22: Ergebnis der Verwendung des Makros `\SinglePicture`

`PlainFigure`

Eine andere Variante zur Erzeugung von Abbildungen innerhalb einer wissenschaftlichen Arbeit stellt die Umgebung `PlainFigure` dar. Diese Umgebung bietet dem Anwender beispielsweise die Möglichkeit, Freitext oder auch direkt durch LaTeX erzeugte Abbildungen in seiner Abhandlung zu integrieren. Dabei erfolgt die Übergabe der einzelnen Parameter analog zur Parameterübergabe des Makros `\SinglePicture`. Hiervon ausgenommen ist natürlich die Verwendung eines Dateinamens, da die Umgebung nicht vorrangig zum Laden externer Bilddateien gedacht ist. Möglich wäre es dennoch.

Da alle Parameter der Umgebung `PlainFigure`, mit Ausnahme der Angabe des Dateinamens, sowohl in ihrer Reihenfolge als auch in den Standardwerten der

beiden optionalen Parameter identisch zum Makro `\SinglePicture` sind, soll an dieser Stelle auf deren ausführliche Erläuterung verzichtet werden. Stattdessen erscheint eine beispielhafte Verwendung der Umgebung `PlainFigure` besser geeignet zu sein.

Beispiel zur Verwendung der Umgebung `PlainFigure`

Angenommen, ein Autor möchte einem Leser den Graphen einer quadratischen Funktion präsentieren, dann kann hierfür die Umgebung `PlainFigure`, wie in Quelltext 16 gezeigt, verwendet werden.

Quelltext 16: Beispiel zur Verwendung der Umgebung `PlainFigure`

```
\begin{PlainFigure}[0.6]% 60% der Textbreite, ohne Rahmen
                   {Parabel der quadratischen Funktion}% Beschriftung
                   {fig:Parabel}%                        Referenzlabel
                   {Quelle: \url{http://www.matheass.de/}}% Quelle
  \centering\setlength{\unitlength}{10mm}
  \begin{picture}(4.3,3.6)(-1.92,-0.35)
  % X-Achse
  \put(-2,0){\vector(1,0){4.4}}\put(2.2,-.3){$x$}
  % Y-Achse
  \put(0,-.4){\vector(0,1){3.6}}\put(-.18,3.08){\makebox(0,0){$y$}}
  % Kurve zeichnen
  \linethickness{.3mm}
  \qbezier(0.0,0.0)(1.2384,0.0)(2.0,2.7622)
  \qbezier(0.0,0.0)(-1.2384,0.0)(-2.0,2.7622)
  \end{picture}
\end{PlainFigure}
```

Nach Erstellung des Dokuments, das den Code aus Quelltext 16 enthält, stellt sich das Ergebnis wie in Abbildung 23 gezeigt dar.

Quelle: http://www.matheass.de/

Abbildung 2: Parabel der quadratischen Funktion

Abbildung 23: Ergebnis der Verwendung der Umgebung `PlainFigure`

`\DoublePicture` und `\DoubleSubPicture`

In wissenschaftlichen Abhandlungen kommt es zuweilen vor, dass Abbildungen zum Beispiel zum Vergleich gegenübergestellt werden müssen. Zur Einbindung solcher Doppelabbildungen kann der Endanwender die Makros `\DoublePicture` und `\DoubleSubPicture` miteinander kombinieren. Doch bevor tiefer in die Materie der Verwendung dieser beiden Makros eingestiegen werden kann, sind noch einige allgemeine Worte zum Hintergrund der durchgeführten Implementierung unerlässlich.

Summiert man alle relevanten Werte, die zur Beschreibung einer Abbildung mit zwei Unterabbildungen vonnöten sind, dann kommt man zu dem Ergebnis, dass hierfür insgesamt zehn Variablen erforderlich wären. Und hierbei wurden nur die wichtigsten Parameter, bestehend aus

- drei Beschriftungen,
- drei Referenzbezeichner,
- zwei Bilddateien und
- zwei Quellenangaben

berücksichtigt. In diesem Zusammenhang sei daran erinnert, dass LaTeX-Makros über maximal neun Parameter verfügen dürfen.

Wie bereits auf Seite 32 angesprochen, kann durch einen Trick die Limitierung der Anzahl der Parameter umgangen werden. Jedoch ist es durchaus fraglich, ob eine derartige Umgehung hilfreich für den Endanwender sei. Denn würde man als Beispiel ein einzelnes Makro mit den oben genannten zehn verpflichtenden Argumenten ausstatten, dann hätte dies sicherlich eine verstärkte Unübersichtlichkeit zur Folge, die unmöglich im Interesse des Anwenders liegen kann!

Daher wurde während der Umsetzung zur Unterstützung von Abbildungen mit Unterabbildungen entschieden, die zugehörige Implementierung in zwei getrennten Makros durchzuführen. Auch wenn die unabhängige Betrachtung der beiden Makros `\DoublePicture` und `\DoubleSubPicture` möglich wäre, gehören sie dennoch zusammen. Denn jedes Makro für sich betrachtet, bietet einem Endanwender wenig bis keinen Nutzen.

Das Makro `\DoublePicture` bildet durch seine vier obligatorischen Parameter `{`⟨*Hauptbeschriftung*⟩`}`, `{`⟨*Hauptreferenzlabel*⟩`}`, `{`⟨*linke Abbildung*⟩`}` und `{`⟨*rechte Abbildung*⟩`}` den notwendigen Rahmen zur Integration von Doppelabbildungen in wissenschaftlichen Abhandlungen. Es verfügt ebenfalls über ein optionales Argument `[box]`, welches die Erzeugung eines separaten Rahmens um jede der beiden Abbildungen erlaubt.

Abweichend von den weiter oben beschriebenen Abbildungskommandos verfügt das Makro `\DoublePicture` über keinen Parameter, der eine Größenanpassung der beiden dargestellten Bilder gestatten würde. Denn beide in den Parametern `{`⟨*linke Abbildung*⟩`}` und `{`⟨*rechte Abbildung*⟩`}` enthaltenen Unterabbildungen werden automatisch skaliert. Dabei erfolgt die Größenanpassung der gesamten Abbildungsbreite auf 90 % und die Skalierung beider Abbildungshälften auf 42 % der vorhandenen Breite des umgebenden Textes. Die sich hieraus ergebende Differenz bildet den Abbildungszwischenraum.

Die Definition der linken beziehungsweise rechten Unterabbildung obliegt dem Makro `\DoubleSubPicture`, das über die vier obligatorischen Parameter `{`⟨*Unterbeschriftung*⟩`}`, `{`⟨*Unterreferenzlabel*⟩`}`, `{`⟨*Unterbilddatei*⟩`}` und `{`⟨*Unterquellenangabe*⟩`}` verfügt. Dieses Makro stellt dem Anwender jedoch keine optionalen Argumente zur Verfügung!

Mit Rücksicht auf den Leser dieser Abhandlung sollte hier nun besser auf die umfassende und detaillierte Behandlung jedes einzelnen Pflichtparameters verzichtet werden, da deren grundsätzliche Bedeutung identisch mit den analogen Argumenten der Kommandos `\SinglePicture` und `PlainFigure` sind.

Des Weiteren sollten die hier gewählten Benennungen den Zweck des jeweiligen Argumentes bereits erkennen lassen. Zudem erscheint die beispielhafte Verwendung der Makros `\DoublePicture` und `\DoubleSubPicture` als ein geeigneteres Mittel, um das Zusammenspiel der hier vorgestellten Makros zu demonstrieren.

Beispiel zur Verwendung von `\DoublePicture` und `\DoubleSubPicture`

Man stelle sich vor, der Autor einer wissenschaftlichen Abhandlung möchte seinem Lesern anhand einer Gegenüberstellung den Unterschied zwischen den verschiedenen Chipkartenarten verdeutlichen. Zu diesem Zweck würde sich die Kombination der `\DoublePicture` und `\DoubleSubPicture` hervorragend eignen, da hierfür nur das in Quelltext 17 gezeigte Codefragment erforderlich wird.

Quelltext 17: Beispiel zur Verwendung des Makros \DoublePicture

```
\DoublePicture[box]% inklusive Rahmen
              {Arten von Chipkarten}% Hauptbeschriftung
              {fig:ChipkartenArten}%  Hauptreferenzlabel
              {\DoubleSubPicture% Unterabbildung links
                      {Aufbau einer Speicherkarte}% Beschriftung
                      {fig:Speicherkarte}%          Referenzlabel
                      {Speicherkarte.pdf}%          Bilddateiname
                      {Quelle: \url{http://www.hardwareecke.de/}}}%Quelle
              {\DoubleSubPicture% Unterabbildung rechts
                      {Aufbau einer Prozessorkarte}% Beschriftung
                      {fig:Prozessorkarte}%          Referenzlabel
                      {Prozessorkarte.pdf}%          Bilddateiname
                      {Quelle: \url{http://www.hardwareecke.de/}}}%Quelle
```

Wurde das Dokument, welches den Code aus Quelltext 17 verwendet, kompiliert, dann enthält die Ausgabe das in Abbildung 24 dargestellte Ergebnis.

Quelle: `http://www.hardwareecke.de/`

(a) Aufbau einer Speicherkarte

Quelle: `http://www.hardwareecke.de/`

(b) Aufbau einer Prozessorkarte

Abbildung 3: Arten von Chipkarten

Abbildung 24: Ergebnis der Verwendung des Makros `\DoublePicture`

`Table`

Hin und wieder kommt es vor, dass Tabellen einer wissenschaftlichen Arbeit nicht standardgerecht formatiert werden. Daher stellt die Datei `fomfod.sty` dem Anwender die Umgebung `Table` zur Verfügung, mit deren Hilfe eine den allgemein anerkannten Empfehlungen genügende Formatierung der Tabellen vorgenommen werden kann.

Wie sich der geneigte Leser eventuell vorstellen kann, ist eine Abstraktion der vielfältigen Varianten existierender Tabellen eine nahezu unlösbare Aufgabe. Aus diesem Grund kann und muss sich der Aufgabenbereich einer Umgebung zur allgemeinen Tabellenkonfiguration auf grundlegende Aktionen beschränken. Hierzu zählen vor allem die standardkonforme Positionierung der Tabellenbeschriftung sowie eine den Richtlinien entsprechende Platzierung der Quellenangabe, falls diese notwendig wird.

Diese Beschränkungen spiegeln sich auch in der Anzahl der benötigten Argumente wider. Denn zur Konfiguration der Umgebung `Table` werden den Anwendern nur die obligatorischen Parameter `{`⟨*Beschriftung*⟩`}`, `{`⟨*Referenzlabel*⟩`}` und `{`⟨*Quellenangabe*⟩`}` bereitgestellt. Eine Angabe optionaler Argumente ist hier nicht erforderlich.

Auch in diesem Teilabschnitt sollte besser auf die ausführliche Beschreibung der einzelnen Parameter verzichtet und stattdessen lieber ein Beispiel zur Darstellung der Zusammenhänge herangezogen werden.

Beispiel zur Verwendung der Umgebung `Table`

Das Beispiel in Quelltext 18 soll zeigen, mit welcher Leichtigkeit die standardkonforme Formatierung der Tabellen einer wissenschaftlichen Abhandlung gelingen kann, wenn hierfür die Umgebung `Table` zum Einsatz kommt.

Quelltext 18: Beispiel zur Verwendung der Umgebung `Table`

```
\begin{Table}% Umgebungsbeginn
      {Übertragungsprotokolle für Chipkarten}% Beschriftung
      {tab:ChipProto}%                         Referenzlabel
      {Quelle: \url{http://www.hardwareecke.de/}}% Quelle
  \begin{tabular}{ll}\hline
    \textbf{Protokoll} & \textbf{Beschreibung}\\\hline
    T = 0  & Asynchron, half-duplex, Byte orientiert\\
    T = 1  & Asynchron, half-duplex, Block orientiert\\
    T = 2  & Asynchron, full-duplex, Block orientiert\\
    T = 3  & Full duplex, wird momentan nicht verwendet\\
    T = 4  & Asynchron, half-duplex, Byte orientiert\\
    T = 14 & für nationale Funktionen\\\hline
  \end{tabular}
\end{Table}% Umgebungsende
```

Das Ausgabeergebnis des in Quelltext 18 enthaltenen Codefragments stellt sich nach einer Kompilierung des entsprechenden Dokuments, wie in Abbildung 25 gezeigt, dar.

Tabelle 1: Übertragungsprotokolle für Chipkarten

Protokoll	Beschreibung
T = 0	Asynchron, half-duplex, Byte orientiert
T = 1	Asynchron, half-duplex, Block orientiert
T = 2	Asynchron, full-duplex, Block orientiert
T = 3	Full duplex, wird momentan nicht verwendet
T = 4	Asynchron, half-duplex, Byte orientiert
T = 14	für nationale Funktionen

Quelle: `http://www.hardwareecke.de/`

Abbildung 25: Ergebnis der Verwendung der Umgebung `Table`

`CodeListing` *und* `\CodeListingFile`

Den letzten Zweig der durch fomfod bereitgestellten Anweisungen zur Formatierung von Gleitobjekten stellen die Kommandos zur Integration von Quellcodelistings dar. Da sowohl die Umgebung `CodeListing` als auch das Makro `\CodeListingFile` die ihnen übergebenen Parameter auf die gleiche Weise verarbeiten, empfiehlt sich auch deren gemeinsame Behandlung. Denn abgesehen von der abweichenden Anwendung von Umgebungen und Makros besteht der einzige Unterschied dieser Befehle in der Bekanntgabe des angezeigten Inhalts.

Die Umgebung `CodeListing` unterstützt zur Konfiguration einer Quelltextüberschrift den obligatorischen Parameter `{`⟨*Beschriftung*⟩`}` und zur Definition eines Bezeichners für Querverweise das Pflichtargument `{`⟨*Referenzlabel*⟩`}`. Eine Anpassung der linken und rechten Quelltexteinrückung ist ebenfalls möglich. Dies kann dann durch den freiwilligen Parameter `[`⟨*Einzug*⟩`]` bewirkt werden, dessen Standardwert auf sieben Punkte voreingestellt wird.

Naturgemäß dienen LaTeX-Umgebungen zur Kapselung von Textpassagen, die in der Regel noch einer speziellen Formatierung unterzogen werden. Solche Textstellen sind vom Grundsatz her eher kürzer gefasst als der umliegende Fließtext. Dieses Grundkonzept lässt auch den bevorzugten Anwendungsfall der hier vorgestellten Umgebung `CodeListing` erkennen. Denn diese Umgebung ist primär zur Erzeugung relativ kurzer Quelltextpassagen gedacht. Die Formatierung längerer Quellcodes ist dennoch möglich.

Auf dem Fachgebiet der Informatik erstellte Arbeiten enthalten nicht selten den Quellcode eines bearbeiteten Softwareprojektes. Die Integration dieser zumeist externen Dateien ist Aufgabe des Makros `\CodeListingFile`. Dieses Makro verfügt, neben dem optionalen Parameter `[`⟨*Einzug*⟩`]` und den obligatorischen Argumenten `{`⟨*Beschriftung*⟩`}` und `{`⟨*Referenzlabel*⟩`}`, zusätzlich auch über den Pflichtparameter `{`⟨*Quellcodedatei*⟩`}`, der zur Bekanntgabe des Namens der einzubindenden Quelltextdatei verwendet wird.

Ein Beispiel soll an dieser Stelle helfen, die marginalen Unterschiede der beiden Kommandos `CodeListing` und `\CodeListingFile` deutlicher herauszuheben und ebenso die überwiegenden Gemeinsamkeiten besser darzustellen.

Beispiel zur Verwendung von `CodeListing` und `\CodeListingFile`

Angenommen, ein Autor möchte den Programmcode des allseits beliebten und oft zitierten Programms „hello, world!“ in seiner Ausarbeitung einbeziehen, da hieran beispielsweise die Grundzüge der Programmiersprache „C“ erläutert werden sollen, dann ist hierfür die Umgebung `CodeListing` bestens geeignet. Quelltext 19 zeigt das zur Umsetzung dieses Szenarios nötige Codefragment.

Quelltext 19: Beispiel zur Verwendung der Umgebung `CodeListing`

```
\begin{CodeListing}[35mm]%              Einzug (links/rechts)
      {Das Programm 'hello, world!'}%  Beschriftung
      {lst:hello}%                      Referenzlabel
  #include <stdio.h>
  int main( void ) {
      printf( "hello, world!\n" );
      return 0;
  }
\end{CodeListing}
```

Eventuell stellt der oben angenommene Autor fest, dass andauernde Programmänderungen eine ständige Aktualisierung der Ausarbeitung zur Folge haben.

Daher beschließt selbiger Autor, die hiervon betroffenen Dateien besser in das Dokument zu importierten. Hierfür ist das Makro `\CodeListingFile` hervorragend geeignet. Quelltext 20 veranschaulicht das zum Import der Beispieldatei `hello.c` notwendige Codefragment.

Quelltext 20: Beispiel zur Verwendung des Makros `\CodeListingFile`

```
\CodeListingFile[35mm]%                 Einzug (links/rechts)
      {Das Programm 'hello, world!'}% Beschriftung
      {lst:hello}%                     Referenzlabel
      {hello.c}%                       Quellcodedatei
```

Unter der Prämisse, dass der Inhalt der in Quelltext 20 verwendeten Beispieldatei mit dem in Quelltext 19 verwendeten Programmcode übereinstimmt, würde sich nach einer Erstellung des entsprechenden Dokuments die in Abbildung 26 dargestellte Ausgabe ergeben.

Listing 1: Das Programm 'hello, world!'

```
#include <stdio.h>
int main( void ) {
    printf( "hello, world!\n" );
    return 0;
}
```

Abbildung 26: Ergebnisse der Verwendung von `CodeListing` und `\CodeListingFile`

3.2.6 Vermischtes

Ein Sammelbecken für weitere nützliche und hilfreiche Makros stellt die *FOM Miscellaneous Definition* dar, die in der Datei `fomisc.sty` implementiert ist. Hierin lassen sich alle Makros finden, die vom Sinnzusammenhang keiner anderen Datei des Pakets fomsdt zugeordnet werden konnten.

Alle bereitgestellten Makros sind so definiert, dass hierfür die Angabe von Dateiladeoptionen nicht erforderlich wird und somit deren Verwendung unabhängig von etwaigen Randbedingungen erfolgen kann.

`\Raise`

Grundsätzlich kann hochgestellter Text mit Hilfe des mathematischen Satzes erzeugt werden, wie das Beispiel $E = m \cdot c^2$ demonstriert. Jedoch führt dessen Verwendung nicht immer zum erwarteten Ergebnis, da beispielsweise hochgestellte Ziffern und hochgestellte Buchstaben unterschiedlich dargestellt werden.

Diese Unterschiede könnten auf den Leser einer wissenschaftlichen Arbeit durchaus irritierend wirken. Daher wird zur verbesserten Darstellung hochgestellter Texte die Verwendung des Makros `\Raise` empfohlen.

Zur Positionierung des im obligatorischen Parameter `{`⟨*hoch*⟩`}` übergebenen Textes wird dieser im ersten Schritt um 85 Prozent der Höhe eines „x“ nach oben verschoben und anschließend auf 70 Prozent der Größe des umliegenden Textes geschrumpft. Dadurch wird sichergestellt, dass bei einer Verwendung der LaTeX-eigenen Makros zur Schriftgrößenänderung der durch `\Raise` erzeugte Hochtext ebenfalls seine Größe anpasst, wie das Beispiel in Quelltext 21 eindrucksvoll demonstriert.

Quelltext 21: Beispiele zur Verwendung des Makros `\Raise`

```
{\scriptsize    Normal\Raise{hoch}}\\
{\footnotesize Normal\Raise{hoch}}\\
{\small         Normal\Raise{hoch}}\\
{\normalsize    Normal\Raise{hoch}}\\
{\large         Normal\Raise{hoch}}\\
{\Large         Normal\Raise{hoch}}\\
{\LARGE         Normal\Raise{hoch}}\\
{\huge          Normal\Raise{hoch}}\\
```

Die gedruckte Version des in Quelltext 21 verwendeten Codefragments stellt sich dann wie in Abbildung 27 gezeigt dar.

Normal[hoch]
Normal[hoch]
Normal[hoch]
Normal[hoch]
Normal[hoch]
Normal[hoch]
Normal[hoch]
Normal[hoch]

Abbildung 27: Ergebnis der Verwendung des Makros `\Raise`

`\TodayShort`

Normalerweise wird das aktuelle Datum innerhalb eines LaTeX-Dokuments durch Verwendung des Makros `\today` eingefügt. Dieses Makro birgt leider den Nachteil, dass der erzeugte Datumstext den ausgeschriebenen, also wörtlich wiedergegebenen Monatsnamen enthält. Das kann unter Umständen hinderlich

sein, wenn als Beispiel die Monatsbezeichnung in Form von Ziffern bevorzugt wird. Aus diesem Grund stellt fomisc das Makro `\TodayShort` zur Verfügung.

Dieses Makro verfügt über zwei optionale Parameter, die zur Modifikation des ausgegebenen Datumstextes verwendet werden können. Hierbei wird der erste optionale Parameter zur grundsätzlichen Vorauswahl des Formates verwendet und der zweite freiwillige Parameter dient der Definition des zu verwendenden Trennzeichens. Sowohl Datumsformat als auch Trennzeichen selektieren ihre Anzeigeinformationen anhand verschiedener Schlüsselwörter, die nun kurz vorgestellt werden sollen.

Erlaubte Schlüsselwörter des ersten Parameters [⟨*Format*⟩]

Wird das Makro `\TodayShort` mit `[din]` als erstem Parameter aufgerufen, dann wird zur Ausgabe des aktuellen Datums das Format *dd.mm.yyyy* verwendet. Diese Formatierung entspricht den Empfehlungen der Norm DIN 5008 und ist auch der Standardwert, wenn keine Angabe des ersten Parameters erfolgt.

Wird hingegen das Makro `\TodayShort` mit `[iso]` als erstem Parameter aufgerufen, dann wird zur Ausgabe des aktuellen Datums das Format *yyyy-mm-dd* verwendet. Diese Formatierung entspricht dann den Empfehlungen der Normen ISO 8601:2004 und EN 28601.

Erlaubte Schlüsselwörter des zweiten Parameters [⟨*Trennzeichen*⟩]

Wird `[def]` dem Makro `\TodayShort` als zweiter Parameter übergeben, dann bewirkt dies die Verwendung des Standardtrennzeichens für Datumsangaben. Dabei erfolgt die Auswahl des Trennzeichens in Abhängigkeit von der Formatangabe des ersten Parameters und ist

- ein Punkt für den Fall einer DIN-gerechten Formatierung und
- ein Gedankenstrich im Falle der ISO-konformen Formatierung!

Wird dem Makro `\TodayShort` hingegen ein [⟨*Freitext*⟩] als zweiter Parameter übergeben, dann wird das angegebene Zeichen beziehungsweise die verwendete Zeichenkette zur Trennung der einzelnen Bestandteile des Datums verwendet. Die Angabe eines „leeren“ Zeichens ist ebenfalls möglich.

Die verschiedenen Möglichkeiten der Kombination der Parameter [⟨*Format*⟩] und [⟨*Trennzeichen*⟩] zeigen die Beispiel in Tabelle 3.

Weitere Makros der Datei `fomisc.sty`

Autoren wissenschaftlicher Arbeiten beziehen sich häufig auf Produkte und Marken von Herstellern, die mit Vermerken wie „Registered“, „Copyright“ oder

auch „Trademark“ ausgestattet sind. Hierfür werden in der Regel spezielle Symbole verwendet, die von fomisc ebenfalls bereitgestellt werden. Beispiele zur Verwendung der Makros `\Registered`, `\Copyright` und `\Trademark` sind

- Google™,
- Microsoft® und
- Copyright © 2007.

Das Makro `\Axel`, das eine Ausgabe der Form AxEl erzeugt, soll den Anwendern des Paket fomsdt die Möglichkeiten des LaTeX-Systems näher bringen. Es ist als Beispiel zur Formatierung von Texten gedacht, deren einzelne Buchstaben „aus der Reihe tanzen“. Andere Beispiele derartiger Texte sind die Eigennamen TeX und LaTeX. Hierdurch soll auch erreicht werden, dass die Anwender des Pakets fomsdt inspiriert werden, eigene Versuche zur Textgestaltung zu unternehmen und spielerisch mit dem LaTeX-System umzugehen.

Tabelle 3: Mögliche Verwendungsformen das Makros `\TodayShort`

Aufrufparameter	Ausgabe	Bemerkung
`\TodayShort`	01.08.2007	DIN-gerecht, Standardtrennung
`\TodayShort[din]`	01.08.2007	DIN-gerecht, Standardtrennung
`\TodayShort[din][def]`	01.08.2007	DIN-gerecht, Standardtrennung
`\TodayShort[din][]`	01082007	DIN-gerecht, ohne Trennung
`\TodayShort[din][/]`	01/08/2007	DIN-gerecht, amerikanische Trennung
`\TodayShort[iso]`	2007-08-01	ISO-konform, Standardtrennung
`\TodayShort[iso][def]`	2007-08-01	ISO-konform, Standardtrennung
`\TodayShort[iso][]`	20070801	ISO-konform, ohne Trennung
`\TodayShort[iso][/]`	2007/08/01	ISO-konform, amerikanische Trennung

3.2.7 Indexkonfiguration

Abweichend zu allen zuvor behandelten Paketelementen verfügt die *FOM Index Style Definition*, implementiert in der Datei `fomidx.ist`, nicht über Makros im klassischen Sinne. Denn sie dient, wie der Dateiname bereits vermuten lässt, ausschließlich der Konfiguration des Layouts des Indexes. Daher wird sich dieser Abschnitt darauf beschränken, einen Überblick über die vorgenommenen Layoutanpassungen des Indexes zu verschaffen.

Basis für die Konfiguration des Indexes bildet die Datei `din.ist`, deren Aufgabe in erster Linie darin besteht, die DIN-gerechte Sortierreihenfolge für Wörter mit deutschen Umlauten festzulegen. Jedoch stellte sich während der Umsetzung heraus, dass diese Sortierung nicht in allen Fällen einwandfrei funktioniert! Somit wurde die Programmierung eines Patches erforderlich, bei der eine Korrektur des LaTeX-Makros `\@wrindex` erfolgte. Da die Datei `fomidx.ist`, wie eingangs bereits erwähnt, keine Makros enthält, wurde die durchgeführte Fehlerbehebung mit in der Datei `fomdes.sty` aufgenommen. Dennoch gehört dieser Patch zur Datei `fomidx.ist` und ist somit der Indexkonfiguration zuzurechnen.

Formatierung der Überschriften (Schriftgröße, Schriftart und Position)
Einleitung der Untereinträge
Indikatoren für Folgeseiten
B
Bildschirm .... 8 ff.
–, Röhre .... 18 f.
Abstand vor den Überschriften
Indikator für Seitenbereiche
M
Monitor .... 8 – 12
Abstand hinter den Überschriften
Einfügen einer Punktfüllung

Abbildung 28: Anpassungen und Änderungen des Layouts des Indexes

Wie Abbildung 28 erkennen lässt, wurden am vordefinierten Aussehen des Stichwortverzeichnisses verschiedentliche Einstellungen und Anpassungen vorgenommen. Dabei orientieren sich alle Änderungen an einer im deutschen Sprachraum oft verwendeten gestalterischen Ausprägung des Indexlayouts, auf die nun näher eingegangen werden soll.

Gruppierung nach Anfangsbuchstaben

Als erste maßgebliche Änderung war eine Gliederung der einzelnen Einträge des Indexes erforderlich. Hierfür wurde als Überschrift der Anfangsbuchstabe der jeweiligen Gruppe gewählt und so formatiert, dass deren Zugehörigkeit deutlich erkennbar wird. Im Einzelnen besteht die Formatierung der Gruppenüberschrift

aus einem links ausgerichteten Großbuchstaben, der fett und ohne Serifen ausgegeben wird. Weiterhin wird eine größere Distanz zur vorhergehenden Gruppe eingestellt, als das für den Abstand zu den eigenen Einträgen der Fall ist.

Abtrennung der Seitenangabe

Im nächsten Schritt wurde die Seitenreferenz eines jeden Indexeintrages rechtsbündig ausgerichtet und der hierdurch entstandene Zwischenraum mit Punkten aufgefüllt. Außerdem wurde die Folgeseitenreferenzierung hinter der Seitenzahl an die im deutschsprachigen Raum typischerweise zu diesem Zweck verwendeten Abkürzungen angepasst. Dabei besteht für die jeweiligen Seitenzahlformatierungen der hier dargelegte Zusammenhang.

… ⟨*n*⟩ f. Indexeinträge, die auf den Seiten ⟨*n*⟩ und ⟨*n+1*⟩ vorkommen, wird die Abkürzung *f.* für „folgende“ angehängt.

… ⟨*n*⟩ ff. Indexeinträge, die auf den Seiten zwischen ⟨*n*⟩ und ⟨*n+2*⟩ vorkommen, wird die Abkürzung *ff.* für „fortfolgende“ angehängt.

… ⟨*n*⟩ – ⟨*m*⟩ Indexeinträge, die sich über mehr als drei Seiten erstrecken, verwenden einen Gedankenstrich, der durch zwei Leerzeichen eingefasst wird.

Einrückung der Untereinträge

Die Standardeinstellungen des Indexes rücken alle Untereinträge eines Haupteintrages um 20 Punkt und alle Untereinträge eines Untereintrages um insgesamt 30 Punkt ein [3]. Dieser Einzug vermittelt nicht selten den Eindruck einer gewissen Unübersichtlichkeit. Daher wurde die Einzugformatierung für Untereinträge angepasst und etwas übersichtlicher gestaltet. Hierfür wird dann die erste Unterebene nicht mehr eingezogen und zu dessen Hervorhebung ein Gedankenstrich vorangestellt, der von einem Komma und einem Leerzeichen gefolgt wird. Die zweite Ebene hingegen wird nur noch um 15 Punkt eingezogen und ebenfalls durch einen Gedankenstrich, gefolgt von Komma und Leerzeichen, eingeleitet. Weitere Unterebenen sind nicht zu berücksichtigen, da sowohl LaTeX als auch das Programm `makeindex` nur bis zu drei Ebenen im Index unterstützt [17, Seite 3].

Abschließend soll noch ein besonderes Merkmal der Datei `fomidx.ist` hervorgehoben werden. Es besteht die Möglichkeit, die Indexkonfiguration auch völlig losgelöst vom Paket fomsdt zu verwenden! Das bedeutet, dass die Anwender, die sich sich eventuell dazu entscheiden, das Paket fomsdt nicht zu verwenden, dennoch die Chance haben, den Index mit einem ansprechenderen Layout zu erzeugen. Der einzige Nachteil entsteht dadurch, dass in diesem Fall der oben

angesprochene Patch nicht aktiv ist und somit alle mit einem Umlaut beginnenden Wörter im Abschnitt „Symbole“ des generierten Indexes einsortiert werden. Dies wäre dann weniger schön.

3.3 Benutzerdokumentation

Ein weiterer wichtiger Bestandteil der Implementierung des Pakets fomsdt war die Bereitstellung einer entsprechenden Benutzerdokumentation, auf die in diesem Abschnitt näher eingegangen werden soll.

Im Fokus des zu erstellenden Benutzerhandbuchs stand, dass es lediglich den Anspruch erheben sollte, dem Endanwender den Umgang mit dem Paket fomsdt zu erläutern. Daher erfolgte die Aufteilung beziehungsweise der Aufbau der einzelnen Handbuchkapitel auf Basis des pragmatischen Ansatzes: *Was interessiert einen Benutzer zu welchem Zeitpunkt.*

Somit wurde während des Verfassens der Paketbeschreibung besonderes Augenmerk auf das Vorhandensein von Anleitungen und Beispielen zur Verwendung der einzelnen Komponenten des Pakets gelegt, da diese die Autoren wissenschaftlicher Arbeiten am ehesten interessieren würden.

Im Gegenzug wurde auf eine LaTeX-konforme Dokumentation aller Quellcodes verzichtet. Dies bedeutet aber nicht, dass auf die Quelltextkommentierung gänzlich verzichtet wurde. Denn für diejenigen Endbenutzer, die selbst Änderungen oder Anpassungen am Paket fomsdt vornehmen möchten, befindet sich eine vollständige Kommentierung des Quelltextes in der zur Auslieferung verwendeten Paketbasis.

Neben einer Beschreibung der Funktionsweisen der einzelnen bereitgestellten Kommandos ist eine ausführliche Anleitung zur Installation des Pakets fomsdt ebenfalls Bestandteil der Benutzerdokumentation.

Die Entscheidung, alle Anweisungen zur Installation in der Dokumentation zu integrieren, wurde unter anderem auf Grund der eingeschränkten Möglichkeiten zur Hervorhebung und Formatierung von Texten innerhalb von ASCII-Dateien getroffen.

Hieraus resultiert der durchaus als positiv zu bewertende Nebeneffekt, dass die Paketdokumentation durch den Endbenutzer erst erzeugt werden muss, bevor mit der eigentlichen Installation begonnen werden kann. Dies gewährleistet, zumindest gilt das für den Durchschnittsbenutzer, dass die Endbenutzerdokumentation auch auf dem Zielsystem vorhanden ist. Folglich kann der ebenfalls zur Dokumentation zu zählende Inhalt der Datei `readme.txt` bis auf eine Kurzanleitung zur Erstellung des Benutzerhandbuches reduziert werden.

3.4 Installationsmethode

Ungeklärt war ebenfalls die Frage nach der Methode der Installation des Pakets fomsdt auf dem Zielsystem. Diese Entscheidung fiel jedoch leicht, da in LaTeX allgemeine Gepflogenheiten [25, 28] im Bezug auf eine Paketinstallation gelten.

Diese Empfehlungen raten an, alle Paketdateien zu einer einzelnen Datei, einer so genannten Documented-TeX-Datei, zusammenzufassen. Weiterhin wird das Beifügen eines Installationsskripts sowie einer Installationsanweisung in Form einer Textdatei befürwortet. Somit besteht die Auslieferungsversion des Pakets fomsdt nur noch aus den Dateien

- `fomsdt.dtx` – das zusammengefasste Paket inklusive Dokumentation,
- `fomsdt.ins` – das zur Installation benötigte Skript und
- `readme.txt` – die Installationsanweisung für den Endbenutzer.

Die Erzeugung der Dateien `fomsdt.dtx` und `fomsdt.ins` erfolgte unter Verwendung des Pakets makedtx, das in der aktuellen Version 0.94b vorlag. Dieses Paket ermöglicht die automatisierte Erstellung der Documented-TeX-Datei `fomsdt.dtx`. Weiterhin kann das benötigte Installationsskript `fomsdt.ins` ebenfalls automatisch durch dieses Paket generiert werden.

Abbildung 29: Erstellprozess der Paketdateien zur Auslieferung

Hingegen wurde die Datei readme.txt, also die Anweisung zur Installation des Pakets fomsdt, händisch erzeugt und dem Distributionspaket durch einfaches Kopieren beigefügt. Diese Datei beinhaltet jedoch lediglich die Anweisungen zur Erzeugung des Benutzerhandbuchs.

Neben der Reduzierung der Anzahl der Dateien, die für eine Distribution zur Verfügung gestellt werden müssen, bietet diese Form der Paketbereitstellung einen weiteren großen Vorteil. Es ist nicht nur möglich, sondern auch sehr empfehlenswert, das Benutzerhandbuch in der Documented-TEX-Datei, also in der zusammengefassten Paketdatei, unterzubringen.

Dieses für den reinen Endanwender des Pakets fomsdt bestimmte Handbuch, das in der in Abbildung 29 gezeigten Datei fomman.tex enthalten ist, wird ebenfalls automatisch in der Datei fomsdt.dtx eingefügt.

Die zuvor angeführten Vorteile bergen jedoch einen inhärenten und nicht zu vernachlässigenden Nachteil! Sowohl die Wahl einer eigenständigen Dokumentenklasse als auch die verwendete Art der Paketdistribution erhöhen die an den Endbenutzer gestellten Anforderungen. Denn zur Durchführung einer Paketinstallation sind nunmehr mehrere Schritte unumgänglich, bei denen es sich im Einzelnen um

- die Erzeugung der Paketdokumentation durch den Endbenutzer,
- das daran anschließende Entpacken des gesamten Pakets,
- das Kopieren der zuvor entpackten Dateien an ihren Bestimmungsort und
- die händische Aktualisierung des LATEX-Grundsystems

handelt. Dennoch überwiegen die Vorteile, wie beispielsweise das Verbergen der inhärenten Komplexität, die durch die gewählte Paketstruktur erreicht werden.

3.5 Verträglichkeit

Nach Abschluss der Implementierungsphase wurde das Paket fomsdt durchgängig getestet. Hierbei wurde besonderer Wert auf die Verträglichkeit des Pakets zu den anderen Klassen und Paketen des LATEX-Systems gelegt. Alle Fehler, die während der Tests auftraten, wurden behoben und daran anschließend der geänderte Code erneut getestet. Probleme, die nicht umgangen oder behoben werden konnten, werden in diesem Abschnitt dokumentiert.

Test mit anderen Dokumentenklassen

Während der durchgeführten Testphase wurden insbesondere die Wechselwirkungen des Pakets fomsdt zu den gängigen Dokumentenklassen article, book,

report, letter und beamer untersucht. Hierbei konnte festgestellt werden, dass die einzelnen Paketkomponenten problemlos mit den meisten LaTeX-eigenen Dokumentenklassen zusammenarbeiten. Im Einzelnen konnten dabei die im Folgenden zusammengefassten Zusammenhänge ermittelt werden.

article Die Dokumentenklasse article bildet die Basis der neu erstellten Klasse fomsdt. Somit wurden alle zugehörigen Komponenten im Umfeld dieser Basisklasse entwickelt. Ein unabhängig durchgeführter Test bestätigte ebenfalls ein problemloses Zusammenwirken der einzelnen Bestandteile des Pakets mit der Klasse article.

book Die einzelnen Komponenten des Pakets fomsdt arbeiten problemlos mit der Dokumentenklasse book zusammen.

report Ebenfalls problemlos arbeiten die einzelnen Komponenten des Pakets fomsdt mit der Dokumentenklasse report zusammen.

letter Die Klasse letter verursachte zusammen mit der Datei `fomfod.sty` diverse leichte bis schwere Fehlermeldungen! Speziell trat hierbei der Fehler `LaTeX Error: Environment figure undefined` auf, der als durchaus ernst zunehmendes Problem zu betrachten ist. Dieser Fehler tritt auch dann auf, wenn die Klasse letter zusammen mit der Umgebung `figure` verwendet wird, ohne dass die Datei `fomfod.sty` darin involviert ist.

Auf Grund dieses Problems wurde entschieden, die Verwendung der Datei `fomfod.sty` zusammen mit der Klasse letter gänzlich zu unterdrücken und stattdessen den Fehler `Package fomfod Error: This package cannot be used with document class 'letter'!` auszugeben und daran anschließend einen Abbruch zu initiieren.

beamer Auch die Dokumentenklasse beamer verursachte diverse Fehlermeldungen, wenn diese zusammen mit der Datei `fomfod.sty` verwendet wurde. Als Ursache dieser Fehler konnte das Paket subfig, welches durch das Makro `\DoublePicture` benötigt wird, ausgemacht werden. Da die aufgetretenen Fehler umgangen werden konnten, ist die Verwendung der Klasse beamer zusammen mit der Datei `fomfod.sty` auch weiterhin möglich. Leider kann in diesem Fall das Makro `\DoublePicture` nicht verwendet werden!

Auch wenn grundsätzlich die Möglichkeit einer Verwendung der einzelnen Komponenten des Pakets fomsdt gemeinsam mit anderen LaTeX-Klassen besteht,

wird dennoch von einer derartigen Anwendung abgeraten! Nicht zuletzt deshalb, weil in diesem Fall die allgemeinen Grundeinstellungen nicht verfügbar sind und die Benutzung der Titelseite, des Sperrvermerks und der ehrenwörtlichen Erklärung ausgeschlossen ist.

Test mit anderen Paketen

Bedingt durch den großen Umfang an verfügbaren LaTeX-Paketen kann sich der geneigte Leser sicherlich vorstellen, dass ein allumfassender Test das Pakets fomsdt in der Tat unrealistisch ist. Daher konnten die einzelnen Tests nur für ausgewählte Pakete durchgeführte werden! Tabelle 4 zeigt die Übersicht der Pakete, mit denen fomsdt problemlos verwendet werden kann.

Tabelle 4: Übersicht der getesteten Pakete

a4wide	gloss	listings	rotating
caption	graphicx	makeidx	subfig
color	hyperref	multicol	tabularx
fancyhdr	ifpdf	natbib	textfit
fontenc	ifthen	ngerman	textpos
footmisc	inputenc	nomencl	ulem

Zusammenfassend kann an dieser Stelle festgehalten werden, dass alle durchgeführten Tests ein hohes Maß an Robustheit des Pakets fomsdt bestätigt haben. Es sind daher nur wenige Ausnahmen bekannt, in denen die Komponenten des Pakets fomsdt anders als vermutlich zu erwarten wäre, agieren. Diese Ausnahmen sollen Gegenstand der nun anschließenden Betrachtungen sein.

Formatierung der Folgeseiten im Index

Wie bereits aus Abbildung 28 ersichtlich wurde, werden im Stichwortverzeichnis die Angaben der Folgeseiten gemäß der Schemata $\langle n \rangle$ *f.*, $\langle n \rangle$ *ff.* sowie $\langle n \rangle - \langle m \rangle$ formatiert. Diese Art der Darstellung birgt den Nachteil, dass die durch das Paket hyperref erzeugten Querverweise innerhalb eines PDF-Dokuments nicht wie erwartet angesprungen werden können.

In anderen Worten, ein Leser, der beispielsweise im Index einer Abhandlung den Verweis *11 ff.* findet, kann durch Klicken der angegebenen Zahl nicht wie erwartet auf *Seite 11* der Abhandlung wechseln. Stattdessen wird zum Anfang der Abhandlung gesprungen. Dieser Umstand ist sicherlich bedauerlich, kann aber auch nicht umgangen werden.

Skalierung von Abbildungen

Versucht der Anwender, die Größe einer mit Hilfe des Makros `\SinglePicture` oder unter Verwendung der Umgebung `PlainFigure` erstellten Abbildung zu manipulieren, dann kann durch die Angabe eines zu klein gewählten Skalierungsfaktors unter Umständen die Platzierung der Elemente Abbildung, Quellenangabe und Beschriftung durcheinander geraten. Dieser Effekt tritt unabhängig von der Textlänge für Quellenangabe und Beschriftung auf und ist somit nur vom Wert des Skalierungsfaktors der Abbildung abhängig. Die genauen Grenzen zeigt die Übersicht in Tabelle 5.

Tabelle 5: Empfohlene Skalierungsuntergrenzen für Abbildungen

Kommando	**Skalierungsfaktor**	**Bemerkung**
`PlainFigure`	0,50	mit umgebenden Rahmen
	0,50	ohne zusätzlichen Rahmen
`\SinglePicture`	0,49	mit umgebenden Rahmen
	0,50	ohne zusätzlichen Rahmen

Dieser Effekt kann freilich vernachlässigt werden, da die Darstellung von zu klein skalierten Abbildungen ohnehin nicht empfehlenswert ist. Denn hierdurch wird die Lesbarkeit des Abbildungsinhaltes negativ beeinflusst.

Doppelbenennung des Makros `\PrintIndex`

Das in der Datei `fomdes.sty` definierte Makro `\PrintIndex` wird in der Datei `doc.sty` unter gleichem Namen definiert. Somit ist dieses Makro nach der Installation des Pakets fomsdt im LaTeX-System an zwei unterschiedlichen Stellen vorhanden!

Durch diese Unverträglichkeit sind dennoch kaum Beeinträchtigungen zu erwarten, da die Datei `doc.sty` in der Regel nicht zur Erstellung wissenschaftlicher Arbeiten verwendet wird!

Kapitelüberschriften und die Option `nopagebreak`

Die Empfehlungen zur Erstellung wissenschaftlicher Arbeiten sagen aus, dass Hauptkapitel auf einer neuen Seite beginnen sollten. Kommt ein Anwender zum Beispiel auf die Idee, hierfür die Definition des Makros `\section`, wie in Quelltext 22 gezeigt, abzuändern, dann kann dies in den Dateien `fomdss.sty` und `fomdes.sty` zu einem Konflikt während der Verarbeitung der Option `nopagebreak` führen!

Quelltext 22: Konflikt mit der Option nopagebreak

```
% Speichere Originaldefinition
\let\tempsection\section
% Erzwinge Seitenvorschub vor jedem Hauptkapitel
\renewcommand\section{\clearpage\tempsection}
```

Dieser Konflikt tritt besonders dadurch hervor, dass trotz der Verwendung der Option `nopagebreak` jedes der erstellten Verzeichnisse auf einer neuen Seite beginnt. Das kann für den Benutzer durchaus ärgerlich sein, da dieses Problem nicht offensichtlich zu erkennen ist!

Falls ein Anwender dennoch nicht auf die Umdekoration des Makros `\section` verzichten will, dann zeigt Quelltext 23 die hierfür anwendbare Umgehung des hier beschriebenen Konflikts.

Quelltext 23: Umgehung des Konflikts der Option nopagebreak

```
\documentclass{fomsdt}
\usepackage[...,nopagebreak]{fomdss}
\usepackage[...,nopagebreak]{fomdes}
\begin{document}
% Verzeichnisse ausgeben
\PrintContents
\PrintAbbrev
\PrintFigures
\PrintTables
% Kapitelüberschriften anpassen.
\let\tempsection\section
\renewcommand\section{\clearpage\tempsection}
% Abhandlungsinhalt
\section{Einleitung}
\section{Grundlagen}
\section{...}
\section{Fazit}
% Kapitelüberschriften zurücksetzen.
\let\section\tempsection
% Verzeichnisse ausgeben
\PrintListings
\PrintGlossary{Glossar}
\PrintBibliography{Literatur}
\PrintIndex
\end{document}
```

Verwendung des Makros `\AtEndDocument`

Verwendet der Anwender das LaTeX-Makro `\AtEndDocument` erst im Anschluss an eine Bekanntgabe der Dokumentenklasse via Makro `\documentclass`, dann tritt bei gleichzeitiger Aktivierung der Eidesstattlichen Versicherung der Effekt auf, dass selbige nicht mehr wie vorgeschrieben auf der letzten Abhandlungsseite zu finden ist!

Diese Phänomen lässt sich dadurch erklären, dass die einzelnen Dateien eines LaTeX-Dokuments sequenziell eingelesen und verarbeitet werden! Da die Klassendatei `fomsdt.cls` das Makro `\AtEndDocument` bereits zur Erzeugung der

Eidesstattlichen Versicherung verwendet, entsteht durch dessen erneute Anwendung ein Konflikt in der Reihenfolge der zuletzt ausgegebenen Seiten. Somit kann dieses Problem nur dann umgangen werden, wenn das Makro `\AtEndDocument` noch vor der Definition der Dokumentenklasse, also vor einem Aufruf des Makros `\documentclass`, erfolgt!

3.6 Kritikpunkte

Dieser Abschnitt enthält Informationen und Anmerkungen zu Punkten, die während der Umsetzung des Pakets fomsdt nicht berücksichtigt werden konnten, aber dennoch die Leser interessieren könnten. Primär soll hierin diskutiert werden, warum einige Implementierungsbestandeile von allgemeingültigen Standards abweichen oder auch warum auf eine Realisierung bestimmter Bereiche gänzlich verzichtet wurde.

Kommandobenennung

Die Erfahrung lehrt, dass die Namensfindung für Befehle und Funktionen, im Fall von LaTeX also die Benennung der Makros und Umgebungen, eine durchaus anspruchsvolle Aufgabe darstellt. Aus diesem Grund verfügt zum Beispiel die Programmiersprache C++ über einen als „namespace“ bezeichneten Bereichsoperator, der vorrangig zur Gruppierung von Klassen und Methoden verwendet wird. Das hehre Ziel derartigen Unterteilungen ist die Definition einzigartiger Kommandonamen, um mögliche Konflikte durch bereits vorhandene Bezeichner abzuschwächen beziehungsweise sie ganz zu unterdrücken.

Leider ist eine so ausgeprägte Gruppierung in LaTeX unmöglich. Daher behilft man sich in der Praxis durch Voranstellen eines Kürzels vor dem eigentlichen Bezeichner. Dies geschieht in der Hoffnung, dass niemand jemals auf die Idee kommt, denselben Bezeichner für seine eigenen Makros zu verwenden. Dieser Ansatz ist grundsätzlich auch sinnvoll, da er das Kernproblem eines Namenskonflikts deutlich reduziert!

In diesem Punkt bietet das Paket fomsdt Anlass zur Kritik, da keinem der veröffentlichten Makros und Umgebungen ein entsprechendes Präfix vorangestellt wurde! Dass dies ein unglücklicher Umstand ist, zeigt sich daran, dass bereits heute ein Namenskonflikt bekannt ist. Dieser Konflikt besteht in der Benennung des Makros `\PrintIndex`, weil die Datei `doc.sty` ebenfalls ein Makro unter diesem Namen definiert. Jedoch ist zu erwarten, dass diese Doppelbenennung nicht weiter problematisch ist, da die Datei `doc.sty` in der Regel nur im Zusammenhang mit der Erstellung einer Paketbeschreibung Verwendung findet. Somit ist die Wahrscheinlichkeit gering, dass ein Benutzer des Pakets fomsdt jemals auf dieses Problem stoßen wird.

Dessen ungeachtet wurde gleichwohl auf die Verwendung eines Kürzels verzichtet, weil zur Verbesserung der Lesbarkeit der einzelnen fomsdt-Bezeichner die Namensgebung durchgängig Groß- und Kleinbuchstaben kombiniert. Dies weicht von der in LaTeX üblichen Benennung ab und bietet somit zumindest einen minimalen Schutz vor möglichen Namensüberschneidungen.

Mehrsprachigkeit

Nach Durchführung einer intensiven Analyse aller notwendigen Aufwände sowie des sich daraus ergebenden Mehrwerts für den Endanwender wurde letztendlich der Entschluss gefasst, im Paket fomsdt auf die Unterstützung verschiedener Sprachen gänzlich zu verzichten! Einer der Gründe, die schlussendlich zu dieser Entscheidung beitrugen, ist der hierfür erforderliche Implementierungsaufwand.

Zudem richtet sich das Paket fomsdt in erster Linie an die Studierenden, die im deutschen Sprachraum ansässig sind. Daher wird davon ausgegangen, dass mit Hilfe des *Pakets zur Erstellung wissenschaftlicher Arbeiten* vorwiegend deutschsprachige Abhandlungen verfasst werden. Dies lässt dann auch einen überdurchschnittlichen Implementierungsaufwand als ungerechtfertigt erscheinen.

Sollte fomsdt dennoch einmal dazu verwendet werden, um eine Arbeit in einer von Deutsch verschiedenen Sprache zu erstellen, dann bleibt einem Anwender immer noch die Möglichkeit, unter Verwendung der jeweiligen Makroparameter alle vordefinierten deutschsprachigen Texte anzupassen.

Quellcodedokumentation

In LaTeX existieren gewisse Gepflogenheiten bezüglich der Dokumentation der Quelltexte. Diese allgemein anerkannten Regeln zielen grundsätzlich darauf ab, den verschiedenen Intensionen der Betrachtung eines Codefragments gerecht zu werden. Dabei wird in erster Linie darauf Wert gelegt, die Implementierung selbst einem Dritten anschaulich und verständlich darzubieten. Das bedeutet auch, dass eine typische LaTeX-Quellcodedokumentation in der Regel die einzelnen Bestandteile der Umsetzung erläutert und einem Anwender somit wenig hilfreiche Informationen zur Verwendung des Codes bietet.

Die fomsdt-Dokumentation weicht von diesem Grundsatz ab! Der hierfür vornehmliche Grund bestand darin, den Endanwendern ein ausführliches und leicht verständliches Benutzerhandbuch mit möglichst vielen Beispielen sowie diversen Tipps und Tricks zur Verfügung zu stellen und ihnen nicht die Beschreibung aller Implementierungsdetails zuzumuten.

Wenn ein Anwender dennoch den Wunsch verspürt, sich die Funktionsweise der einzelnen Makros näher anschauen zu wollen, dann steht ihm immer noch die Option offen, in der Datei `fomsdt.dtx` den vollständig dokumentierten Quellcode einzusehen.

Mathematischer Satz

Das Setzen mathematischer Formeln bleibt innerhalb des Pakets fomsdt gänzlich unbeachtet! Denn gegen die entsprechende Unterstützung sprach vor allem eine nahezu unmöglich durchzuführende Abstraktion der mannigfaltigen Varianten mathematischer Zusammenhänge zu einer allgemeingültigen Struktur.

Auf der anderen Seite ist die Einbeziehung mathematischer Formeln in einem LaTeX-Dokument derart einfach, dass jeglicher hierfür zu leistender Implementierungsaufwand zuzüglich der notwendigen Tests sowie einer Dokumentation des Verhaltens den daraus erwachsenden Nutzen deutlich übersteigen würde!

Zur Demonstration dieser Einfachheit der Verwendung mathematischer Formeln in LaTeX-Dokumenten soll an dieser Stelle das Beispiel in Quelltext 24 dienlich sein. Dieses Beispiel zeigt unter anderem die wenigen notwendigen Handgriffe, um alle im Text verwendeten Formeln einheitlich zu definieren und diese dann auch in einem entsprechenden Verzeichnis am Dokumentenbeginn auszugeben.

Quelltext 24: Beispiel zur Demonstration der Einfachheit der Einbeziehung mathematischer Formeln in ein LaTeX-Dokument

```
% Die Formelnummerierung am linken Rand verwenden!
\documentclass[leqno]{fomsdt}
% Präambel
\makeatletter
\def\equationshead{Formelverzeichnis}% Text der Überschrift
\def\listofequations{% Verzeichniserzeugung
  \section*{\equationshead}%   Kapitelüberschrift ausgeben
  % sowie dessen Eintragung im Inhaltsverzeichnis vornehmen
  \addcontentsline{toc}{section}{\equationshead}%
  \@starttoc{loe}}
\makeatother

\newcommand{\formula}[3]{%     Kommando zur einfachen
  \begin{equation}%            Formelverwaltung definieren
  % Übergebenen Beschriftung im Formelverzeichnis eintragen
  \addcontentsline{loe}{subsection}{\theequation\qquad#1}%
  \label{#2}#3%       Label definieren und Formel ausgeben
  \end{equation}}

% Text
\begin{document}
\listofequations% Formelverzeichnis ausgeben
% Formeln definieren
\formula{Die erste Binomische Formel}{binom1}{(a+b)^2=a^2+2ab+b^2}
\formula{Die Äquivalenz von Masse und Energie}{emc2}{E=m\cdot c^2}
% Formeln referenzieren
Die erste Binomische Formel in Formel~\ref{binom1} hat nichts mit
der Massen"~Energie"~Äquivalenz aus Formel~\ref{emc2} gemeinsam!
\end{document}
```

4 Schlussbetrachtung

Die vorliegende Diplomarbeit zeigt, dass das Verfassen einer wissenschaftlichen Abhandlung bisweilen recht komplizierte Züge annehmen kann. Und dies ist nicht zuletzt dadurch bedingt, dass von vielen Studierenden die Einhaltung diverser Regularien mit zum Teil widersinnigen und auch widersprüchlichen Vorschriften abverlangt wird. Das soll aber nicht heißen, dass Vorschriften und Regeln grundsätzlich schlecht oder gar falsch sind! Nur wäre ein „gesundes" Maß an Auflagen, unter Beachtung allgemeiner Grundsätze zur Verbesserung der Lesbarkeit für alle Autoren von Seminar- und Abschlussarbeiten ebenso wünschenswert wie hilfreich. Jedoch konnten die vorherrschenden Verhältnisse im Rahmen dieser Abhandlung nur festgestellt werden.

Andererseits zeigt diese Abschlussarbeit auch, dass eine Verwendung von LaTeX den Autoren während der Erstellung ihrer wissenschaftlichen Arbeiten in verschiedenen Bereichen helfen kann. Sei es beispielsweise durch die automatisierte Erzeugung von Verzeichnissen oder auch durch eine vorlagengestützte Erstellung des Titelblattes. Dennoch sollte nicht vergessenen werden, dass trotz großer praktischer Erfahrung das Schreiben eines Dokuments in LaTeX mehr Zeit in Anspruch nimmt, als dies vergleichsweise durch den Einsatz eines der gängigen Textverarbeitungssysteme der Fall ist. Gleichwohl rechtfertigt das ausgegebene Endergebnis diesen Mehraufwand.

Beide zuvor reflektierten Aspekte, also die Gestaltung der Schriftstücke in Bezug auf eine verbesserte Lesbarkeit sowie die vereinfachte Anwendung von LaTeX, versucht das Paket fomsdt weitestgehend miteinander zu vereinen. Aus diesem Grund wurde auch auf eine Implementierung abstruser Regelungen so gut es ging verzichtet; einzige Ausnahme bildet die Option `toctotoc`, deren Aufgabe darin besteht, die Überschrift des Inhaltsverzeichnisses ebenfalls mit in das Inhaltsverzeichnis aufzunehmen. Zugleich bietet das *Paket zur Erstellung wissenschaftlicher Arbeiten* allen Anwendern genügend Flexibilität, um auch zukünftig das Anfertigen von Seminar- und Abschlussarbeiten zu ermöglichen.

Ein Benutzerhandbuch mit vielen Anwendungsbeispielen ist ebenfalls Bestandteil des Pakets fomsdt. Erst dieses Handbuch erlaubt dem Anwender die einfache Handhabung der einzelnen Komponenten des Pakets. Dabei ist der Aufbau der Dokumentation in weiten Teilen einem „Kochbuch" nachempfunden und enthält daher vorwiegend die Erklärungen, *wie* ein bestimmtes Kommando verwendet werden sollte. Im Gegenzug wurde bewusst auf die funktionale Beschreibung der einzelnen Kommandos verzichtet, da ausladende Details der Implementierung in einem Handbuch mit Fokus auf eine Verwendung in der Regel eher hinderlich sind und somit nicht tatsächlich zur Übersichtlichkeit beitragen.

Bei der Wahl der Installationsmethode wurde besonderes Augenmerk darauf gelegt, dass sich das *Paket zur Erstellung wissenschaftlicher Arbeiten* auf mög-

lichst vielen Plattformen gleichermaßen einrichten lässt. Diese Randbedingung hat nun aber zur Folge, dass die Integration des Pakets fomsdt in einer vorhandenen LaTeX-Umgebung verschiedene Aktionen des Anwenders unumgänglich werden lässt. Sicherlich wäre eine Optimierung des Verfahrens zur Installation ebenso denkbar wie wünschenswert, jedoch stellt eine Rücksichtnahme auf alle Details der verfügbaren LaTeX-Plattformen ein nicht zu vernachlässigendes Mehr an Aufwand dar. Somit bietet die aktuell unterstützte Installationsvariante ein ausgewogenes Maß an Komfort und Flexibilität.

Technologiebetrachtungen

Eine der herausragenden Eigenschaften von LaTeX ist dessen Verfügbarkeit für diverse Betriebssysteme. Dieser Umstand ermöglicht allen LaTeX-Anwendern eine plattformenübergreifende Erstellung von Schriftsätzen. Dies stellt einen großen Vorteil dar, wenn sich beispielsweise ein Autor veranlasst sieht, eine Abhandlung auf verschiedenen Rechnern mit unterschiedlichen Betriebssystemen verfassen zu müssen. In diesem Fall können die Quellen des Dokuments dann einfach von einem System auf ein anderes kopiert und anschließend das Ausgabedokument problemlos erstellt werden. Das vom jeweiligen LaTeX-Kern produzierte Ergebnis wird in beiden Fällen eine identische Brillanz aufweisen.

Eine ebenfalls bestechende Eigenschaft des LaTeX-Systems ist die Tatsache, dass alle Dokumentquellen aus einfachen ASCII-Dateien bestehen und somit zu deren Bearbeitung ein simpler Texteditor völlig ausreicht. Dies ist als gravierender Vorteil gegenüber gängigen Textverarbeitungsprogrammen zu werten, da diese nicht selten zu Programmabstürzen neigen, je umfangreicher und größer die damit erstellten Dokumente werden.

An dieser Stelle darf auch der Hinweis nicht fehlen, dass sämtliche zur Erstellung von LaTeX-Dokumenten notwendigen Programme als kostenlose Downloads im Internet verfügbar sind. Das gilt nicht nur für den LaTeX-Kern sondern auch für alle Pakete. Auch existieren viele kostenlose Programme, die zur Bearbeitung einfacher Textdateien hervorragend geeignet sind.

Selbst wenn das Textsatzsystem LaTeX den Anwendern ein im Vergleich zu üblichen Textverarbeitungen überragendes Ausgabeergebnis liefert, sollten freilich einige unschöne Perspektiven nicht unterschlagen werden.

Vom Standpunkt eines Paketentwicklers aus betrachtet stellt das Fehlen eines Verfahrens zur Privatisierung von Kommandos unter Umständen ein größeres Problem dar. Denn Pakete enthalten nicht selten Algorithmen, die nur für den internen Einsatz gedacht sind. Da nun die Möglichkeit besteht, auch intern verwendete Variablen oder Makros und Umgebungen von außerhalb eines Pakets zu manipulieren, ohne hierfür den Originalcode ändern zu müssen, wird hiervon auch des Öfteren Gebrauch gemacht. Dies kann zwar zur Lösung eines speziellen Problems oder zur Behebung eines Fehlers, wie im Falle der Korrektur des

Makros `\@wrindex` geschehen, sehr nützlich sein, ist dessen ungeachtet aber als Schwachstelle des Systems zu bezeichnen, da zum Beispiel durch die fehlerhafte Umdekoration eines Kommandos ein unvorhersagbares Ausgabeergebnis entstehen kann.

Eine weitere Schwäche von LaTeX ist die durchaus als kompliziert zu bezeichnende Art der Implementierung mehrerer optionaler Kommandoparameter. Hier wäre sicherlich eine prozedurbasierte oder gar klassenorientierte Möglichkeit der Kommandodefinition, wie beispielsweise in der Programmiersprache Java™ verwendet, besser geeignet. Jedoch wird dieser Ansatz wohl auch in ferner Zukunft eher nicht verfolgt werden, da hierfür die Definition einer gänzlich neuen Sprache und somit die komplette Überarbeitung des LaTeX-Kernsystems nötig wäre.

Auch stellt das Fehlen einer Möglichkeit zur Definition des Gültigkeitsbereichs von Kommandos und Variablen für Paketentwickler eine gewisse Schwierigkeit dar. Denn jeder Bezeichner, der in einem Paket definiert wird, dabei spielt eine interne oder externe Verwendung keine Rolle, sollte im gesamten LaTeX-System nur einmal vergeben sein. Anderenfalls besteht die Möglichkeit von Konflikten durch konkurrierende Bezeichner! Diese Problematik spielt in der Praxis aber nur eine untergeordnete Rolle, da in der Regel die Paketentwickler auf diese Einschränkung Rücksicht nehmen. Sollte, wie im Fall des Makros `\PrintIndex`, dennoch ein Konfliktpotenzial vorhanden sein, dann kann dies meistens vernachlässigt werden. Jedoch ist es stets Aufgabe des Entwicklers, die Möglichkeit des jeweiligen Konflikts zu dokumentieren und eventuell auch eine entsprechende Problemumgehung bereitzustellen.

Die Quintessenz des Pakets fomsdt

Einer der Hauptaspekte, die durch das Paket fomsdt positiv ins Gewicht fallen, ist die Zusammenfassung vieler verschiedener Vorschriften zu einer sinnvollen Regeluntermenge, die zur Erstellung wissenschaftlicher Abhandlung wirklich benötigt werden. Hierzu zählen unter anderem die verwendete Schriftgröße, der vorgegebene Zeilenabstand und auch der voreingestellte Seitenrand, der eine lesefreundliche Zeilenbreite zur Folge hat. Alle implementierten Regeln dienen einzig und allein der Verbesserung der Lesbarkeit von Dokumenten, die mit Hilfe des Pakets fomsdt erstellten werden. Nicht mehr aber auch nicht weniger.

Auch sollen durch das Paket fomsdt wenig geübte Endanwender befähigt werden, ihre Abhandlungen in LaTeX zu setzen. Daher wurden typische Elemente wissenschaftlicher Ausarbeitungen soweit abstrahiert, dass die Definition allgemeingültiger Kommandos möglich wurde. Diese Kommandos erlauben dem Anwender des Pakets fomsdt Abbildungen, Tabellen, Codelistings und auch alle Verzeichnisse einheitlich und vor allem regelkonform zur gestalten.

Ebenfalls besteht die Möglichkeit, charakteristische Absatzlayouts auf einfache Art und Weise zu erzeugen. Zu diesem Zweck stellt fomsdt entsprechende Kom-

mandos zur Verfügung, mit deren Hilfe die Gestaltung spezieller Layouts erst praktikabel wird.

Der nächste große Pluspunkt des Pakets fomsdt ist die automatisierte Erstellung des Sperrvermerks sowie der Eidesstattlichen Versicherung. Beide für Abschlussarbeiten unverzichtbaren Elemente werden durch fomsdt im Ausgabedokument an den „richtigen“ Stellen platziert. Ein Anwender braucht sich also hierüber keinerlei Gedanken zu machen.

Unter Einbeziehung aller in dieser Abschlussarbeit aufgeführten positiven und negativen Aspekte ist nunmehr deutlich erkennbar, dass LaTeX mit all seinen technischen Möglichkeiten die perfekte Grundlage zur Erstellung von Seminar- und Abschlussarbeiten bietet. Durch Verwendung des Pakets fomsdt können die bereits vorhandenen Mittel dann noch effektiver eingesetzt werden, somit ist es sehr empfehlenswert, wissenschaftliche Ausarbeitungen mit LaTeX und unter Einsatz des Pakets fomsdt zu erstellen.

A Quellcode des Pakets fomsdt

A.1 Dokumentenklasse (Quelldatei fomsdt.cls)

```
1:  %%------------------------------------------------------------
2:  %
3:  % File:        fomsdt.cls
4:  %
5:  % Description: Definition file for the LaTeX document class
6:  %              "FOM Scientific Document Template".
7:  %
8:  % Author:      Axel Kesseler
9:  %
10: % Version:     1.0.8
11: %
12: % Date:        2007/10/29
13: %
14: %-------------------------------------------------------------
15: \NeedsTeXFormat{LaTeX2e}[1995/12/01]
16: \ProvidesClass{fomsdt}
17:   [2007/10/29 v1.0 FOM Scientific Document Template]
18: %-------------------------------------------------------------
19: % Declaration of all package options.
20: %-------------------------------------------------------------
21: % Option 'moderntitle' enables the usage of title pages with
22: % modern look and feel. A special title page style is NOT
23: % used by default.
24: \newif\if@usetitlemodern
25: \@usetitlemodernfalse
26: \DeclareOption{moderntitle}
27:   {\@usetitlemoderntrue%
28:    \@usetitleclassicfalse}
29: % Option 'classictitle' enables the usage of title pages with
30: % modern look and feel. A special title page style is NOT
31: % used by default.
32: \newif\if@usetitleclassic
33: \@usetitleclassicfalse
34: \DeclareOption{classictitle}
35:   {\@usetitlemodernfalse%
36:    \@usetitleclassictrue}
37: % Option 'coloredlinks' enables the usage of colored links
38: % to highlight cross references within a document. Colored
39: % links are NOT used by default.
40: \newif\if@usecoloredlinks
41: \@usecoloredlinksfalse
42: \DeclareOption{coloredlinks}
43:   {\@usecoloredlinkstrue}
44: % Define a value to store the width of a separation line
45: % between the page heading and the rest of the text. The
46: % default width of this line is set to 0.5pt.
```

```
47:  \newlength{\@headrule}
48:  \setlength{\@headrule}{0.5pt}
49:  % Option 'noheadrule' disables the usage of a separation
50:  % line between page heading and the rest of the text simply
51:  % through reset the value of '@headrule' to 0pt. The heading
52:  % separation is used by default.
53:  \DeclareOption{noheadrule}
54:    {\setlength{\@headrule}{0pt}}
55:  % Option 'nochapter' disables the usage of chapter within
56:  % page headings. The chapter within page headings will be
57:  % used by default.
58:  \newif\if@usechapter
59:  \@usechaptertrue
60:  \DeclareOption{nochapter}
61:    {\@usechapterfalse}
62:  % Option 'lockflag' enables the usage of the "lock flag"
63:  % on the page behind the title page. The "lock flag" is
64:  % NOT used by default.
65:  \newif\if@uselockflag
66:  \@uselockflagfalse
67:  \DeclareOption{lockflag}
68:    {\@uselockflagtrue}
69:  % Option 'affidavit' enables the usage of the affidavit
70:  % on the last page. The affidavit is NOT used by default.
71:  \newif\if@useaffidavit
72:  \@useaffidavitfalse
73:  \DeclareOption{affidavit}
74:    {\@useaffidavittrue}
75:  %-----------------------------------------------------------
76:  % Declare additionally options which are used to configure
77:  % the base class 'article'. All these options are already
78:  % defined within the base class!
79:  %-----------------------------------------------------------
80:  % This IF is used to distinguish between default font size
81:  % and a font size given by the user. The assumption is that
82:  % the user hasn't used this originally article-class option.
83:  \newif\if@defaultfontsize
84:  \@defaultfontsizetrue
85:  % Declare supported font size options.
86:  \DeclareOption{10pt}{%
87:    \@defaultfontsizefalse
88:    \PassOptionsToClass{10pt}{article}}
89:  \DeclareOption{11pt}{%
90:    \@defaultfontsizefalse
91:    \PassOptionsToClass{11pt}{article}}
92:  % This IF is used to distinguish between default page size
93:  % and a page size given by the user. The assumption is that
94:  % the user hasn't used this originally article-class option.
95:  \newif\if@defaultpagesize
96:  \@defaultpagesizetrue
97:  % Declare supported paper size options.
```

```
 98:  \DeclareOption{a5paper}{%
 99:    \@defaultpagesizefalse
100:    \PassOptionsToClass{a5paper}{article}}
101:  \DeclareOption{b5paper}{%
102:    \@defaultpagesizefalse
103:    \PassOptionsToClass{b5paper}{article}}
104:  \DeclareOption{letterpaper}{%
105:    \@defaultpagesizefalse
106:    \PassOptionsToClass{letterpaper}{article}}
107:  \DeclareOption{legalpaper}{%
108:    \@defaultpagesizefalse
109:    \PassOptionsToClass{legalpaper}{article}}
110:  \DeclareOption{executivepaper}{%
111:    \@defaultpagesizefalse
112:    \PassOptionsToClass{executivepaper}{article}}
113:  % Pass all further (unknown for class 'fomsdt') options
114:  % to base class 'article'.
115:  \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
116:  \ProcessOptions\relax
117:  % Use default settings only if necessary!
118:  \if@defaultfontsize
119:    \PassOptionsToClass{12pt}{article}
120:  \fi
121:  \if@defaultpagesize
122:    \PassOptionsToClass{a4paper}{article}
123:  \fi
124:  %-----------------------------------------------------------
125:  % Now load the base class 'article'.
126:  %-----------------------------------------------------------
127:  \ProcessOptions\relax%
128:  \LoadClass{article}
129:  %-----------------------------------------------------------
130:  % Load all depending packages.
131:  %-----------------------------------------------------------
132:  % Load package 'inputenc' with option 'latin1' to specify
133:  % that 'latin1' input encoding should use.
134:  \RequirePackage[latin1]{inputenc}
135:  % Load package 'ngerman' to support german stuff.
136:  \RequirePackage{ngerman}
137:  % Load package 'fontenc' with option 'T1' to specify the
138:  % font encodings to be used.
139:  \RequirePackage[T1]{fontenc}
140:  % Load package 'color' to support the usage of different
141:  % text colors.
142:  \RequirePackage{color}
143:  % Internally used to check macro parameters.
144:  \RequirePackage{ifthen}
145:  % Load package 'a4wide' to support the usage of the whole
146:  % printable area of an A4 page.
147:  \RequirePackage{a4wide}
148:  % Load package 'footmisc' with options 'hang' and 'stable'
```

```
149:   % to support special footnote layout.
150:   \RequirePackage[hang,stable]{footmisc}
151:   % Load package 'fancyhdr' to support different layouts of
152:   % of headings and footers.
153:   \RequirePackage{fancyhdr}
154:   % Used to check the output format.
155:   \RequirePackage{ifpdf}
156:   %-------------------------------------------------------------
157:   % Adjust page size and page margin.
158:   %-------------------------------------------------------------
159:   \setlength{\topmargin}     {0cm}
160:   \setlength{\marginparsep}  {0cm}
161:   \setlength{\marginparwidth}{0cm}
162:   \setlength{\oddsidemargin} {1cm}
163:   \setlength{\evensidemargin}{1cm}
164:   \setlength{\headheight}    {1cm}
165:   \setlength{\headsep}       {0.8cm}
166:   %% REMARK: Set text width BEFORE adjusting the settings
167:   %%         of headings and footer because it's using it!
168:   \setlength{\textwidth}{\paperwidth}
169:   \addtolength{\textwidth}{-35mm}
170:   \addtolength{\textwidth}{-35mm}
171:   %-------------------------------------------------------------
172:   % General paragraph settings.
173:   %-------------------------------------------------------------
174:   \newlength{\default@parskip}
175:   \setlength{\default@parskip}{%
176:     0.5\baselineskip plus 0.5ex minus 0.3ex}
177:   \setlength{\parskip}{\default@parskip}
178:   \setlength{\parindent}{0mm}
179:   \sloppy
180:   %-------------------------------------------------------------
181:   % Settings for footnotes.
182:   %-------------------------------------------------------------
183:   \setlength{\footnotemargin}{2.6ex}
184:   %-------------------------------------------------------------
185:   % Settings for headings and footer.
186:   %-------------------------------------------------------------
187:   \pagestyle{fancy}
188:   \if@usechapter
189:     \renewcommand{\sectionmark}[1]{\markboth{#1}{#1}}
190:   \else
191:     \renewcommand{\sectionmark}[1]{}
192:     \renewcommand{\markboth}[2]{}{}
193:   \fi
194:   \fancyhf{}
195:   \fancyhead[R]{\nouppercase{\rightmark}}
196:   \fancyhead[R]{\nouppercase{\leftmark}}
197:   \fancyfoot[C]{\vspace{1ex}\small{\thepage}}
198:   \renewcommand{\headrulewidth}{\@headrule}
199:   %% ATTENTION: Leave a single line empty
```

```
200:  %%            after the next code line!
201:  \fancypagestyle{plain}
202:
203:  %------------------------------------------------------------
204:  % Clubber and widow settings.
205:  %------------------------------------------------------------
206:  \clubpenalty=10000
207:  \widowpenalty=10000
208:  \displaywidowpenalty=10000
209:  %------------------------------------------------------------
210:  % Definition of lock flag depending macros
211:  %------------------------------------------------------------
212:  \newif\if@siglockflag
213:  \newif\if@fomlockflag
214:  \newif\if@stdlockflag
215:  % If the user sets the option 'lockflag' then the lock flag
216:  % will be generated on a blank page automatically. This page
217:  % directly follows the title page.
218:  \newcommand\print@lockflag{%
219:  % Do not modify current page style if option 'lockflag' is
220:  % NOT used.
221:  \if@uselockflag
222:    \clearpage\thispagestyle{empty}
223:  \fi
224:  % Start with lock flag.
225:  \parbox[b]{\textwidth}{\list{}{%
226:    \setlength{\listparindent}{0em}
227:    \setlength{\itemindent}{\listparindent}
228:    \setlength{\leftmargin}{15mm}
229:    \setlength{\rightmargin}{\leftmargin}
230:    \setlength{\parsep}{\default@parskip}
231:    \setlength{\itemsep}{\parsep}}
232:    \item\textbf{\large\lockflag@header}
233:    \item\lockflag@text
234:    % Print the signature if it should be used.
235:    \if@siglockflag
236:      \par\vspace{6em}
237:      \addtolength{\textwidth}{-2\leftmargin}
238:      \parbox[b]{0.45\textwidth}{%
239:        \raggedright{%
240:          {\lockflag@date}\par%
241:           \footnotesize\lockflag@sigleft}}
242:      \parbox[b]{0.54\textwidth}{%
243:        \raggedleft{%
244:          \footnotesize\lockflag@sigright}}
245:      \addtolength{\textwidth}{2\leftmargin}
246:    \fi
247:  \endlist}%
248:  % Do not use a new page if option 'lockflag' is NOT used.
249:  \if@uselockflag\clearpage\fi}
250:  % Data of this macro will be used within lock flag 'fom'
```

```
251:  % and 'std'.
252:  \newcommand{\LockflagCompany}[1]{%
253:    \def\lockflag@company{#1}}
254:  % Data of this macro will be used within lock flag's
255:  % signature.
256:  \newcommand{\LockflagDate}[1]{\def\lockflag@date{#1}}
257:  % This macro generates lock flag content depending on given
258:  % parameter #2.
259:  % ** If parameter #2 equals 'fom' then the FOM stanard lock
260:  %    flag will be used.
261:  %    -- In this case the macros '\General', '\Title'
262:  %       and '\LockflagCompany' has to be used to set
263:  %       appropriated data beforehand!
264:  % ** If parameter #2 equals 'std' then a general lock flag
265:  %    will be used.
266:  %    -- In this case the macros '\General' and
267:  %       '\LockflagCompany' has to be used to set
268:  %       appropriated data beforehand!
269:  % ** Otherwise given parameter #2 will be printed as it is.
270:  %    -- In this case the macros '\General', '\Title' and
271:  %       '\LockflagCompany' will be ignored within this
272:  %       context!
273:  % If the given optional parameter #1 equals 'sig' then the
274:  % lock flag's signature will also be used.
275:  % ** In this case the macro '\LockflagDate' has to be used
276:  %    to set appropriated data beforehand!
277:  \newcommand\LockflagContent[2][]{%
278:    \ifthenelse{\equal{#1}{sig}}%
279:      {\@siglockflagtrue}%
280:      {\@siglockflagfalse}%
281:    \ifthenelse{\equal{#2}{std}}%
282:      {\@stdlockflagtrue%
283:       \@fomlockflagfalse}%
284:      {\ifthenelse{\equal{#2}{fom}}%
285:        {\@fomlockflagtrue%
286:         \@stdlockflagfalse}%
287:        {\@stdlockflagfalse%
288:         \@fomlockflagfalse}}%
289:    % Prepare general lock flag content.
290:    \if@stdlockflag
291:      \long\def\lockflag@text{Die vorliegende {\@General} %
292:      beinhaltet interne vertrauliche Informationen der %
293:      Firma {\lockflag@company}. Die Weitergabe des Inhalts %
294:      der {\@General} im Gesamten oder in Teilen sowie das %
295:      Anfertigen von Kopien oder Abschriften -- auch in %
296:      digitaler Form -- sind grundsätzlich untersagt. %
297:      Ausnahmen bedürfen der schriftlichen Genehmigung %
298:      der Firma {\lockflag@company}.}%
299:    % Prepare FOM specific lock flag content.
300:    \else\if@fomlockflag
301:      \long\def\lockflag@text{Die vorliegende {\@General} %
```

```
302:       mit dem Titel {\@Title} enthält unternehmensinterne %
303:       Daten der Firma {\lockflag@company}. Daher ist sie %
304:       nur zur Vorlage bei der FOM sowie den Begutachtern %
305:       der Arbeit bestimmt. Für die Öffentlichkeit und %
306:       dritte Personen darf sie nicht zugänglich sein.}%
307:     % Just use data of parameter #2.
308:     \else
309:       \long\def\lockflag@text{#2}
310:   \fi\fi}%
311:   % This macro allows to change lockflag's left signature
312:   % content. The default value for the left signature side
313:   % is set to '(Ort, Datum)'.
314:   \newcommand{\LockflagSigLeft}[1]{%
315:     \def\lockflag@sigleft{#1}}
316:   % This macro allows to change lockflag's right signature
317:   % content. The default value for the right signature side
318:   % is set to '(Eigenhändige Unterschrift)'.
319:   \newcommand{\LockflagSigRight}[1]{%
320:     \def\lockflag@sigright{#1}}
321:   % This macro allows to change the content of lockflag's
322:   % header. The default value of the lockflag header is set
323:   % to 'Sperrvermerk'.
324:   \newcommand{\LockflagHeader}[1]{%
325:     \def\lockflag@header{#1}}
326:   % Set lock flag's defaults to avoid error messages.
327:   \LockflagDate{}
328:   \LockflagCompany{\textbf{\textsl{???}}}
329:   \LockflagContent{std}
330:   \LockflagSigLeft{(Ort, Datum)}
331:   \LockflagSigRight{(Eigenhändige Unterschrift)}
332:   \LockflagHeader{Sperrvermerk}
333:   %-----------------------------------------------------------
334:   % Definition of affidavit depending macros
335:   %-----------------------------------------------------------
336:   \newif\if@sigaffidavit
337:   \newif\if@stdaffidavit
338:   \newcommand\print@affidavit{%
339:     % Do not print the affidavit if the option 'affidavit'
340:     % is not used.
341:     \if@useaffidavit
342:       % Ensure an empty new page.
343:       \clearpage\thispagestyle{empty}
344:       % Start with affidavit.
345:       \parbox[b]{\textwidth}{\list{}{%
346:         \setlength{\listparindent}{0em}
347:         \setlength{\itemindent}{\listparindent}
348:         \setlength{\leftmargin}{15mm}
349:         \setlength{\rightmargin}{\leftmargin}
350:         \setlength{\parsep}{\default@parskip}
351:         \setlength{\itemsep}{\parsep}}
352:         \item\textbf{\large\affidavit@header}
```

```
353:       \item\affidavit@text
354:       % Print the signature if it should be used.
355:       \if@sigaffidavit
356:         \par\vspace{6em}
357:         \addtolength{\textwidth}{-2\leftmargin}
358:         \parbox[b]{0.45\textwidth}{%
359:           \raggedright{%
360:             {\affidavit@date}\par%
361:              \footnotesize\affidavit@sigleft}}
362:         \parbox[b]{0.54\textwidth}{%
363:           \raggedleft{%
364:             \footnotesize\affidavit@sigright}}
365:         \addtolength{\textwidth}{2\leftmargin}
366:       \fi
367:     \endlist}
368:   \fi}
369: % Data of this macro will be used within affidavit's
370: % signature.
371: \newcommand{\AffidavitDate}[1]{\def\affidavit@date{#1}}
372: % This macro generates affidavit's content depending on
373: % given parameter #2.
374: % ** If parameter #2 equals 'std' then a general affidavit
375: %    will be used.
376: %    -- In this case the macro '\General' has to be used to
377: %       set appropriated data beforehand!
378: % ** Otherwise the given parameter #2 will be printed as it
379: %    is.
380: %    -- In this case the macro '\General' will be ignored
381: %       within this context!
382: % If the given optional parameter #1 equals 'sig' then the
383: % affidavit's signature will also be used.
384: % ** In this case the macro '\AffidavitDate' has to be used
385: %    to set appropriated data beforehand!
386: \newcommand\AffidavitContent[2][]{
387:   \ifthenelse{\equal{#1}{sig}}
388:     {\@sigaffidavittrue}
389:     {\@sigaffidavitfalse}
390:   \ifthenelse{\equal{#2}{std}}
391:     {\@stdaffidavittrue}
392:     {\@stdaffidavitfalse}
393:   % Prepare general affidavit content.
394:   \if@stdaffidavit
395:     \long\def\affidavit@text{Hiermit versichere ich, %
396:     dass die vorliegende {\@General} von mir selbstständig %
397:     und ohne unerlaubte Hilfe angefertigt worden ist, %
398:     insbesondere, dass ich alle Stellen, die wörtlich oder %
399:     annähernd wörtlich aus Veröffentlichungen entnommen %
400:     sind, durch Zitate als solche gekennzeichnet habe. %
401:     Ich versichere auch, dass die von mir eingereichte %
402:     schriftliche Version mit der digitalen Version %
403:     übereinstimmt. Weiterhin erkläre ich, dass die %
```

```
404:       {\@General} in gleicher oder ähnlicher Form noch %
405:       keiner anderen Prüfungsbehörde vorgelegen hat.}%
406:     % Just use data of parameter #2.
407:     \else
408:       \long\def\affidavit@text{#2}
409:     \fi}
410:   % This macro allows to change affidavit's left signature
411:   % content. The default value for the left signature side
412:   % is set to '(Ort, Datum)'.
413:   \newcommand{\AffidavitSigLeft}[1]{%
414:     \def\affidavit@sigleft{#1}}
415:   % This macro allows to change affidavit's right signature
416:   % content. The default value for the right signature side
417:   % is set to '(Eigenhändige Unterschrift)'.
418:   \newcommand{\AffidavitSigRight}[1]{%
419:     \def\affidavit@sigright{#1}}
420:   % This macro allows to change the content of affidavit's
421:   % header. The default value of the affidavit header is set
422:   % to 'Ehrenwörtliche Erklärung'.
423:   \newcommand{\AffidavitHeader}[1]{%
424:     \def\affidavit@header{#1}}
425:   % Set affidavit's defaults to avoid error messages.
426:   \AffidavitDate{}
427:   \AffidavitContent{std}
428:   \AffidavitSigLeft{(Ort, Datum)}
429:   \AffidavitSigRight{(Eigenhändige Unterschrift)}
430:   \AffidavitHeader{Ehrenwörtliche Erklärung}
431:   %-----------------------------------------------------------
432:   % PDF document settings.
433:   %-----------------------------------------------------------
434:   \ifpdf
435:     % Load hyperref-package only for PDF output formats.
436:     \RequirePackage[%
437:       colorlinks=true,%
438:       bookmarks=true,%
439:       plainpages=false%
440:     ]{hyperref}%
441:     % Define PDF properties. Every of the following macros
442:     % uses one obligational parameter which changes the
443:     % corresponding PDF property.
444:     \newcommand{\PDFTitle}[1]{%
445:       \hypersetup{pdftitle={#1}}}
446:     \newcommand{\PDFAuthor}[1]{%
447:       \hypersetup{pdfauthor={#1}}}
448:     \newcommand{\PDFSubject}[1]{%
449:       \hypersetup{pdfsubject={#1}}}
450:     \newcommand{\PDFProducer}[1]{%
451:       \hypersetup{pdfproducer={#1}}}
452:     % Set defaults for PDF properties.
453:     \PDFTitle{}
454:     \PDFAuthor{}
```

```
455:     \PDFSubject{}
456:     \PDFProducer{PDFLaTeX}
457:     % Define hyperlink color for the dokument.
458:     \if@usecoloredlinks
459:       \definecolor{linkcolor}{rgb}{0,0,0.75}
460:     \else
461:       \definecolor{linkcolor}{rgb}{0,0,0}
462:     \fi
463:     % Setup hyperlink color for the whole document.
464:     \hypersetup{%
465:       pdfborder         = 0 0 0,
466:       bookmarksopen     = true,
467:       bookmarksnumbered = true,
468:       linkcolor         = linkcolor,
469:       citecolor         = linkcolor,
470:       filecolor         = linkcolor,
471:       urlcolor          = linkcolor}
472:   \else
473:     % Define it to avoid error messages.
474:     \newcommand{\PDFTitle}[1]{#1}
475:     \newcommand{\PDFAuthor}[1]{#1}
476:     \newcommand{\PDFSubject}[1]{#1}
477:     \newcommand{\PDFProducer}[1]{#1}
478:   \fi
479:   %------------------------------------------------------------
480:   % Document's title settings.
481:   %------------------------------------------------------------
482:   \newcommand{\General}[1]{%
483:     \def\@General{#1}}
484:   \newcommand{\Date}[1]{%
485:     \def\@Date{#1}}
486:   \newcommand{\Institute}[1]{%
487:     \def\@Institute{#1}}
488:   \newcommand{\Course}[1]{%
489:     \def\@Course{#1}}
490:   \newcommand{\Semester}[1]{%
491:     \def\@Semester{#1}}
492:   \newcommand{\Subject}[1]{%
493:     \def\@Subject{#1}}
494:   \newcommand{\Title}[1]{%
495:     \def\@Title{#1}}
496:   \newcommand{\Professor}[2][Betreuer:]{%
497:     \def\@ProfessorLabel{#1}\def\@ProfessorName{#2}}
498:   \newcommand{\Student}[2][Autor:]{%
499:     \def\@StudentLabel{#1}\def\@StudentName{#2}}
500:   %------------------------------------------------------------
501:   % Set title default values.
502:   %------------------------------------------------------------
503:   % All values which are not initialized with defaults will
504:   % produce an error message while compilation!
505:   \General  {Hausarbeit}
```

```
506:  \Date     {Essen, den \today}
507:  \Institute{FOM -- Fachhochschule für %
508:             Oekonomie \& Management Essen}
509:  \Course   {Berufsbegleitender Studiengang}
510:  \Semester {}
511:  %------------------------------------------------------------
512:  % Configure title page.
513:  %------------------------------------------------------------
514:  \if@usetitlemodern
515:    % Load further required packages.
516:    \RequirePackage{rotating}
517:    \RequirePackage[overlay]{textpos}
518:    % Overwrite command for titel creation.
519:    \renewcommand\maketitle{%
520:      \begin{titlepage}%
521:        % Reset page margin temporarily.
522:        \setlength{\topmargin}     {0cm}
523:        \setlength{\headheight}    {0cm}
524:        \setlength{\headsep}       {0cm}
525:        \setlength{\oddsidemargin} {0cm}
526:        \setlength{\evensidemargin}{0cm}
527:        \setlength{\marginparsep}  {0cm}
528:        \setlength{\marginparwidth}{0cm}
529:        \setlength{\footskip}      {0cm}
530:        % Save original text height and width!
531:        \newlength{\OLDtextheight}
532:        \newlength{\OLDtextwidth}
533:        \setlength{\OLDtextheight}{\textheight}
534:        \setlength{\OLDtextwidth}{\textwidth}
535:        % Increase text height and width!
536:        \setlength{\textheight}  {\paperheight}
537:        \addtolength{\textheight}{-2in}
538:        \setlength{\textwidth}   {\paperwidth}
539:        \addtolength{\textwidth} {-2in}
540:        % Used for a proper PDF linkage.
541:        \renewcommand*{\thepage}{Title}
542:        % Textbox for rotated left side text.
543:          \begin{textblock*}{0.14\textwidth}(0pt,0pt)%
544:          \textcolor[rgb]{0.8,0.8,0.8}{%
545:            \begin{sideways}%
546:              \parbox{\textheight}{%
547:                \begin{flushright}%
548:                  \usefont{T1}{phv}{bx}{sl}%
549:                    \selectfont\Huge\@General\normalfont
550:                \end{flushright}}%
551:            \end{sideways}%
552:          \rule{1pt}{\textheight}}%
553:        \end{textblock*}%
554:        % Create textbox for main part.
555:        \begin{textblock*}%
556:          {0.88\textwidth}(0.14\textwidth,0pt)%
```

```
557:          \parbox[t][0.86\textheight][c]{\textwidth}{%
558:            \begin{flushright}%
559:                {\huge\bf\@Institute \par}\vspace{18mm}%
560:                {\large\bf\@Course   \par}\vspace{5mm}%
561:                {\bf\@Semester       \par}\vspace{18mm}%
562:                {\Large\bf\@Subject  \par}\vspace{18mm}%
563:                {\Large\bf\@Title    \par}%
564:            \end{flushright}}%
565:        \end{textblock*}%
566:        % Create author textbox below main part.
567:        \begin{textblock*}%
568:          {0.88\textwidth}(0.14\textwidth,0.88\textheight)%
569:        \begin{minipage}[t][0.12\textheight][b]{\textwidth}%
570:          \begin{list}{}{%
571:            \setlength{\leftmargin}{5em}%
572:            \setlength{\labelwidth}{0.9\leftmargin}%
573:            \setlength{\labelsep}{0.1\leftmargin}%
574:            \renewcommand{\makelabel}[1]{##1\hfil}}%
575:          \item[\@ProfessorLabel]\@ProfessorName%
576:          \item[\@StudentLabel]\@StudentName%
577:          \item[\@Date]%
578:          \end{list}%
579:        \end{minipage}%
580:        \end{textblock*}%
581:        % Restore original text height and width!
582:        \setlength{\textheight}{\OLDtextheight}
583:        \setlength{\textwidth}{\OLDtextwidth}
584:      \end{titlepage}
585:      \if@uselockflag
586:        % In this case the page counter has to be adjusted.
587:        % This is necessary to suppress LaTeX-warning
588:        % "destination with the same identifier".
589:        \setcounter{page}{0}
590:        \print@lockflag
591:        \setcounter{page}{1}
592:      \fi
593:    }%
594:  \else\if@usetitleclassic
595:    % Load further required packages.
596:    \RequirePackage[overlay]{textpos}
597:    % Overwrite command for titel creation.
598:    \renewcommand\maketitle{%
599:      \begin{titlepage}%
600:        % Used for a proper PDF linkage.
601:        \renewcommand*{\thepage}{Title}
602:        % Create textbox for main part.
603:        \begin{textblock*}{\textwidth}(0pt,0pt)%
604:          \parbox[t][0.86\textheight][c]{\textwidth}{%
605:            \begin{center}
606:                {\huge\bf\@Institute \par}\vspace{18mm}%
607:                {\large\bf\@Course   \par}\vspace{5mm}%
```

```
608:                {\bf\@Semester       \par}\vspace{18mm}%
609:                {\Large\bf\@Subject  \par}\vspace{18mm}%
610:                {\Large\bf\@Title    \par}%
611:              \end{center}}%
612:            \end{textblock*}%
613:            % Create author textbox below main part.
614:            \begin{textblock*}{\textwidth}(0pt,0.90\textheight)%
615:            \begin{minipage}[t][0.12\textheight][b]{\textwidth}%
616:              \begin{list}{}{%
617:                \setlength{\leftmargin}{4.7em}%
618:                \setlength{\labelwidth}{0.9\leftmargin}%
619:                \setlength{\labelsep}{0.1\leftmargin}%
620:                \renewcommand{\makelabel}[1]{##1\hfil}}%
621:              \item[\@ProfessorLabel]\@ProfessorName%
622:              \item[\@StudentLabel]\@StudentName%
623:              \item[\@Date]%
624:              \end{list}%
625:            \end{minipage}%
626:            \end{textblock*}%
627:          \end{titlepage}
628:          \if@uselockflag
629:            % In this case the page counter has to be adjusted.
630:            % This is necessary to suppress LaTeX-warning
631:            % "destination with the same identifier".
632:            \setcounter{page}{0}
633:            \print@lockflag
634:            \setcounter{page}{1}
635:          \fi
636:        }%
637:      \else
638:        % For scientific documents the usage of the standard
639:        % LaTeX title macros is not recommended. In this case
640:        % print a warnig to inform the user about this.
641:        \ClassWarningNoLine{fomsdt}%
642:          {Using standard LaTeX-Title is not recommended!}
643:      \fi\fi
644:      %------------------------------------------------------------
645:      % Do all jobs at the end of the document.
646:      %------------------------------------------------------------
647:      \AtEndDocument{%
648:        % The affidavit should always be on the last page!
649:        \print@affidavit%
650:      }%
651:      %
652:      %------------------------------------------------------------
653:      % Finish class file.
654:      %%-----------------------------------------------------------
```

A.2 Dokumentenbeginn (Quelldatei `fomdss.sty`)

```
 1: %%-----------------------------------------------------------
 2: %
 3: % File:        fomdss.sty (FOM Document Start Section)
 4: %
 5: % Description: This file defines macros for a document's
 6: %              startup section. A scientific document
 7: %              always includes a table of contents and
 8: %              mostly a list of abbreviations, list of
 9: %              figures and also a list of tables.
10: %
11: % Author:      Axel Kesseler
12: %
13: % Version:     1.0.7
14: %
15: % Date:        2007/10/29
16: %
17: %------------------------------------------------------------
18: \NeedsTeXFormat{LaTeX2e}[1995/12/01]
19: \ProvidesPackage{fomdss}
20:   [2007/10/29 v1.0 FOM Document Start Section]
21: %------------------------------------------------------------
22: % Declaration of all package options.
23: %------------------------------------------------------------
24: % Option 'toctotoc' puts the heading of "table of contents"
25: % (value of \contentsname) into the table of contents itself.
26: % The heading of TOC is NOT part of TOC by default.
27: \newif\if@usetoctotoc
28: \@usetoctotocfalse
29: \DeclareOption{toctotoc}
30:   {\@usetoctotoctrue}
31: % Option 'nopagebreak' avoids the usage of new pages for
32: % every single directory list within the document's start
33: % section. For scientific documents every directory list
34: % should start on a new page. That's the reason why the
35: % page break has to be swiched off explicitly. The page
36: % break is used by default.
37: \newif\if@dssusenewpage
38: \@dssusenewpagetrue
39: \DeclareOption{nopagebreak}
40:   {\@dssusenewpagefalse}
41: % Option 'abbreviation' enables the usage of the "list of
42: % abbreviations" within the document's start section because
43: % the usage of a "list of abbreviations" is NOT used by
44: % default.
45: \newif\if@useabbreviation
46: \@useabbreviationfalse
47: \DeclareOption{abbreviation}
48:   {\@useabbreviationtrue}
49: % Option 'abbrevrefpage' enables the usage of a page
```

```
50:   % reference within "list of abbreviations" because the
51:   % page reference is NOT used by default.
52:   \newif\if@useabbrevrefpage
53:   \@useabbrevrefpagefalse
54:   \DeclareOption{abbrevrefpage}
55:     {\@useabbrevrefpagetrue}
56:   % Print a warning if the package receives an unknown option
57:   % just to inform the user.
58:   \DeclareOption*{\PackageWarningNoLine{fomdss}
59:     {Unknown option '\CurrentOption'}}
60:   \ProcessOptions\relax%
61:   %-----------------------------------------------------------
62:   % Load all depending packages.
63:   %-----------------------------------------------------------
64:   % Used to check the output format.
65:   \RequirePackage{ifpdf}
66:   %-----------------------------------------------------------
67:   % Declarations for table of contents (TOC)
68:   %-----------------------------------------------------------
69:   % Definition of macro which creates the table of contents.
70:   % This macro supports one optional parameters which can be
71:   % used to change the TOC's heading.
72:   % Param1: Represents the heading of the table of contents
73:   %         to be used and its default is set to 'Inhalt'.
74:   \newcommand{\PrintContents}[1][Inhalt]{%
75:     % Set TOC's heading (defualt is 'Inhalt').
76:     \renewcommand{\contentsname}{#1}
77:     % Always force a page break for TOC!
78:     \clearpage%
79:     % Used to fix a bug within hyperef-package.
80:     \ifpdf\phantomsection\fi
81:     % If TOC's heading has to appear within the TOC itself
82:     % then add it to the TOC!
83:     \if@usetoctotoc
84:       \addcontentsline{toc}{section}{\contentsname}
85:     \fi
86:     % Create the tabel of contents.
87:     \tableofcontents%
88:     % Always force a page break for TOC!
89:     \clearpage%
90:   }%
91:   %-----------------------------------------------------------
92:   % Declarations for list of abbreviations (LOA).
93:   %-----------------------------------------------------------
94:   % A list of abbreviations should be used within the current
95:   % document. 'if@useabbreviation' is TRUE when the user loads
96:   % the package with option 'abbreviation'. By default
97:   % 'if@useabbreviation' is FALSE.
98:   \if@useabbreviation
99:     % As mentioned in the package documentation the usage of
100:    % the list of abbreviations produces a message to inform
```

```
101:  % the user about "How to handle the creation of list of
102:  % abbreviations". Just print this user message.
103:  \typeout{%
104:    *************************************************^^J%
105:    * PLEASE NOTE! The usage of fomdss' abbreviations^^J%
106:    * requires the nomencl package! Please ensure that^^J%
107:    * nomencl package is installed on your LaTeX system^^J%
108:    * in version 4.2, dated at 2005/09/22 (or higher).^^J%
109:    * Furthermore you have to run the program 'makeindex'^^J%
110:    * with options:^^J%
111:    * makeindex <file>.nlo -s nomencl.ist -o <file>.nls^^J%
112:    *************************************************}%
113:  % Load package 'nomencl' to support the "list of
114:  % abbreviations" depending on package option
115:  % 'abbrevrefpage'.
116:  \if@useabbrevrefpage
117:    \RequirePackage[german,refpage]{nomencl}
118:  \else
119:    \RequirePackage[german]{nomencl}
120:  \fi
121:  % Load package 'ulem' with option 'normalem' to support
122:  % single underlined text.
123:  \RequirePackage[normalem]{ulem}
124:  % Definition of macro which marks an abbreviation within
125:  % a text to be added to the list of abbreviations. This
126:  % macro defines two obligational parameters. The first
127:  % parameter will be also printed within the text.
128:  % Param1: Represents the abbreviation itself.
129:  % Param2: Represents the explanation of the abbreviation.
130:  % Sample: The following piece of code produces the text
131:  %         fragment "... a PC is used ..." and also adds
132:  %         "PC ... Personal Computer" to the list of
133:  %         abbreviations":
134:  %         ... a \Abbrev{PC}{Personal Computer} is used ...
135:  \newcommand\Abbrev[2]{#1\nomenclature{#1}{#2}}
136:  % Setup the width of abbreviation labels to 20% of
137:  % currently used text width.
138:  \setlength{\nomlabelwidth}{.2\textwidth}
139:  % Reset the layout of abbreviation's label: Used bold font
140:  % and fill the gap between label and explanation with dots.
141:  \renewcommand{\nomlabel}[1]{\textbf{#1} \dotfill}
142:  % Set gap between every abbreviation list entry to 50% of
143:  % currently used paragraph separator.
144:  \setlength{\nomitemsep}{0.5\parsep}
145:  % Definition of macro to highlight abbreviation letters
146:  % within an abbreviation explanation. This macro defines
147:  % one obligational parameter which underlines the enclosed
148:  % text.
149:  % Param1: Represents the abbreviation's letter to be
150:  %         highlighted within an abbreviation's explanation.
151:  % Sample: The following piece of code produces the text
```

```
152:     %         fragment "... a PC is used ..." and also adds
153:     %         "PC ... Personal Computer" with underlined "P"
154:     %         and "C" to the list of abbreviations":
155:     %         ... a \Abbrev{PC}{\Mark{P}ersonal \Mark{C}omputer}
156:     %         is used ...
157:     \newcommand{\Mark}[1]{\uline{#1}}
158:     % The list of abbreviationa has to be activated.
159:     \makenomenclature
160:     % Definition of macro which creates the list of
161:     % abbreviations. This macro supports one optional
162:     % parameter which can be used to change the list of
163:     % abbreviations' heading.
164:     % Param1: Represents the list of abbreviations' heading
165:     %         to be used. The default heading is set to
166:     %         'Abkürzungen'.
167:     \newcommand{\PrintAbbrev}[1][Abkürzungen]{%
168:       % Set list of abbreviations' heading (default is
169:       % 'Abkürzungen').
170:       \renewcommand{\nomname}{#1}
171:       % Force a page break only if it's requested.
172:       \if@dssusenewpage\clearpage\fi
173:       % Used to fix a bug within hyperef-package.
174:       \ifpdf\phantomsection\fi
175:       % Add list of abbreviations' heading to TOC!
176:       \addcontentsline{toc}{section}{\nomname}
177:       % Create the list of abbreviations.
178:       \printnomenclature%
179:       % Reset headings! This command has been placed here
180:       % because of some problems with heading!
181:       \sectionmark{\nomname}
182:       % Force a page break only if it's requested.
183:       \if@dssusenewpage\clearpage\fi
184:     }%
185:   \else
186:     % The macros \Abbrev, \Mark and \PrintAbbrev will be also
187:     % available if the package option 'abbreviation' wasn't
188:     % used. This avoids error messages while compilation!
189:     \newcommand{\Abbrev}[2]{#1}
190:     \newcommand{\Mark}[1]{}
191:     \newcommand{\PrintAbbrev}[1]{}
192:   \fi
193:   %------------------------------------------------------------
194:   % Declarations for list of figures (LOF)
195:   %------------------------------------------------------------
196:   % Definition of macro which creates the list of figures.
197:   % This macro supports one optional parameter which can be
198:   % used to change list of figures' heading. This macro is
199:   % always available and cannot be suppressed by another
200:   % package option.
201:   % Param1: Represents the list of figures' heading to be used.
202:   %         The default heading is set to 'Abbildungen'.
```

```
203:  \newcommand{\PrintFigures}[1][Abbildungen]{%
204:    % Set list of figures' heading (default is 'Abbildungen').
205:    \renewcommand{\listfigurename}{#1}
206:    % Force a page break only if it's requested.
207:    \if@dssusenewpage\clearpage\fi
208:    % Used to fix a bug within hyperef-package.
209:    \ifpdf\phantomsection\fi
210:    % Add list of figures' heading to TOC!
211:    \addcontentsline{toc}{section}{\listfigurename}
212:    % Create list of figures.
213:    \listoffigures%
214:    % Force a page break only if it's requested.
215:    \if@dssusenewpage\clearpage\fi
216:  }%
217:  %-----------------------------------------------------------
218:  % Declarations for list of tables (LOT)
219:  %-----------------------------------------------------------
220:  % Definition of macro which creates the list of tables.
221:  % This macro supports one optional parameter which can be
222:  % used to change list of tables' heading. This macro is
223:  % always available and cannot be suppressed by another
224:  % package option.
225:  % Param1: Represents the list of tables' heading to be used.
226:  %         The default heading is set to 'Tabellen'.
227:  \newcommand{\PrintTables}[1][Tabellen]{%
228:    % Set list of figures' heading (default is 'Tabellen').
229:    \renewcommand{\listtablename}{#1}
230:    % Force a page break only if it's requested.
231:    \if@dssusenewpage\clearpage\fi
232:    % Used to fix a bug within hyperef-package.
233:    \ifpdf\phantomsection\fi
234:    % Add list of tables' heading to TOC!
235:    \addcontentsline{toc}{section}{\listtablename}
236:    % Create list of tables.
237:    \listoftables%
238:    % Force a page break only if it's requested.
239:    \if@dssusenewpage\clearpage\fi
240:  }%
241:  %
242:  %-----------------------------------------------------------
243:  % Finish style file.
244:  %%----------------------------------------------------------
```

A.3 Dokumentenende (Quelldatei `fomdes.sty`)

```
1:  %%----------------------------------------------------------
2:  %
3:  % File:        fomdes.sty (FOM Document End Section)
4:  %
5:  % Description: This file defines macros for a document's
```

```
6:  %              end section. A scientific document mostly
7:  %              includes a glossary, an index, a list of
8:  %              code listings etc.
9:  %
10: % Author:      Axel Kesseler
11: %
12: % Version:     1.0.7
13: %
14: % Date:        2007/10/29
15: %
16: %-----------------------------------------------------------
17: \NeedsTeXFormat{LaTeX2e}[1995/12/01]
18: \ProvidesPackage{fomdes}
19:   [2007/10/29 v1.0 FOM Document End Section]
20: %-----------------------------------------------------------
21: % Declaration of all package options.
22: %-----------------------------------------------------------
23: % Option 'index' enables the usage of an index wihtin the
24: % document's end section because an index is NOT used by
25: % default.
26: \newif\if@useindex
27: \@useindexfalse
28: \DeclareOption{index}
29:   {\@useindextrue}
30: % Option 'listings' enables the usage of the "table of code
31: % listings" within the document's end section because the
32: % table of code listings is NOT used by default.
33: \newif\if@uselistings
34: \@uselistingsfalse
35: \DeclareOption{listings}
36:   {\@uselistingstrue}
37: % Option 'glossary' enables the usage of the glossary within
38: % the document's end section because the usage of a glossary
39: % is NOT used by default.
40: \newif\if@useglossary
41: \@useglossaryfalse
42: \DeclareOption{glossary}
43:   {\@useglossarytrue}
44: % Option 'glossrefpage' enables the usage of a page reference
45: % within the glossary because the page reference is NOT used
46: % by default.
47: \newif\if@useglossrefpage
48: \@useglossrefpagefalse
49: \DeclareOption{glossrefpage}
50:   {\@useglossrefpagetrue}
51: % Option 'nobibliography' avoids a usage of the bibliography
52: % within the document's end section. For scientific documents
53: % the usage of the bibliography is stricly recommended.
54: % That's the reason why the bibliography has to be swiched
55: % off explicitly. The bibliography is used by default.
56: \newif\if@usebibliography
```

```
57:  \@usebibliographytrue
58:  \DeclareOption{nobibliography}
59:    {\@usebibliographyfalse}
60:  % Option ’nopagebreak’ avoids the usage of new pages for
61:  % every single directory list within the document’s end
62:  % section. For scientific documents every directory list
63:  % should start on a new page. That’s the reason why the
64:  % page break has to be swiched off explicitly. The page
65:  % break is used by default.
66:  \newif\if@desusenewpage
67:  \@desusenewpagetrue
68:  \DeclareOption{nopagebreak}
69:    {\@desusenewpagefalse}
70:  % Print a warning if the package receives an unknown option
71:  % just to inform the user.
72:  \DeclareOption*{\PackageWarningNoLine{fomdes}
73:    {Unknown option ’\CurrentOption’}}
74:  \ProcessOptions\relax%
75:  %-----------------------------------------------------------
76:  % Load all depending packages.
77:  %-----------------------------------------------------------
78:  % Used to check the output format.
79:  \RequirePackage{ifpdf}
80:  % Internally used to check macro parameters.
81:  \RequirePackage{ifthen}
82:  %-----------------------------------------------------------
83:  % Index depending settings and definitions.
84:  %-----------------------------------------------------------
85:  % Definition of an if-statement to distinguish between the
86:  % usage of single- or multi-columns for the index. This ’if’
87:  % has to be defined "out of every block"! Otherwise an error
88:  % message comes up while compilation.
89:  \newif\if@useidxcols
90:  % An index should be used within the current document.
91:  % ’if@useindex’ is TRUE when the user loads the package with
92:  % option ’index’. By default ’if@useindex’ is FALSE.
93:  \if@useindex
94:    % As mentioned in the package documentation the usage of
95:    % an index produces a message to inform the user about
96:    % "How to handle the index creation". Just print this user
97:    % message.
98:    \typeout{%
99:      **************************************************^^J%
100:     * PLEASE NOTE! Index creation requires to run:^^J%
101:     * makeindex <file>.idx^^J%
102:     * But it is recommended to create an index based on^^J%
103:     * ISO style. For this run:^^J%
104:     * makeindex <file>.idx -s fomidx.ist^^J%
105:     * Especially for German index creation users should^^J%
106:     * use option ’-g’ to ensure German word ordering^^J%
107:     * rules according to DIN 5007. In this case run:^^J%
```

```
108:       * makeindex <file>.idx -g -s fomidx.ist^^J%
109:       **************************************************}%
110:     % Load package 'multicol' to support multiple index
111:     % columns. In conjunction with the support of "suppress
112:     % page break" the usage of package 'multicol' is the
113:     % "better way" instead the usage of macro '\twocolumn',
114:     % which always performs a page break.
115:     \RequirePackage{multicol}
116:     % Load package 'makeidx' (the core of index creation) and
117:     % also activate the index generation.
118:     \RequirePackage{makeidx}
119:     \makeindex
120:     % Following definitions are imported for words starting
121:     % with german umlauts in conjunction with an index
122:     % creation. Otherwise, words starting with "ä,ö,ü" would
123:     % be inserted in the symbols section of the index.
124:     \newcommand\org@wrindex{}
125:     \let\org@wrindex\@wrindex
126:     \def\@wrindex{%
127:       \edef"{\string"}%
128:       \def Ä{"A}%
129:       \def Ö{"O}%
130:       \def Ü{"U}%
131:       \def ä{"a}%
132:       \def ö{"o}%
133:       \def ü{"u}%
134:       \def ß{"s}%
135:       \org@wrindex}
136:     % Definition of macro which creates the index. This macro
137:     % supports two optional parameters which can be used to
138:     % fine tune the layout of index.
139:     % Param1: Represents the index's heading to be used. The
140:     %         default heading is set to 'Index'.
141:     % Param2: Represents the index's number of columns to be
142:     %         used and its default is set to 'double'. The
143:     %         following pre-defined key words are available
144:     %         to this parameter:
145:     %         single: The index uses only one column.
146:     %         double: The index uses two columns [default]!
147:     %         triple: The index uses three columns.
148:     %         Every different key word given through the
149:     %         second parameter will produce a package waring
150:     %         and results in a usage of the default number of
151:     %         columns!
152:     \newcommand{\PrintIndex}[1][Index]{%
153:       \@ifnextchar[%
154:         {\@iiPrintIndex[#1]}
155:         {\@iiPrintIndex[#1][double]}%]
156:     }
157:     \long\def\@iiPrintIndex[#1][#2]{%
158:       % Set index's heading (defualt is 'Index').
```

```
159:       \renewcommand{\indexname}{#1}
160:       % Re-define the index environment for better parameter
161:       % handling.
162:       \renewenvironment{theindex}{%
163:         % Setup index section name.
164:         \section*{\indexname}
165:         % The Index always belongs to the TOC!
166:         \addcontentsline{toc}{section}{\indexname}
167:         % Setup index heading (if set through the class).
168:         \sectionmark{\indexname}
169:         % Prepare index column setup.
170:         \newcounter{idxcols}
171:         \ifthenelse{\equal{#2}{single}}
172:           % Switch multi-columns off.
173:           {\@useidxcolsfalse}
174:           % Switch multi-columns on.
175:           {\@useidxcolstrue
176:             % Set number of multi-columns.
177:             \ifthenelse{\equal{#2}{double}}
178:             {\setcounter{idxcols}{2}}
179:             {\ifthenelse{\equal{#2}{triple}}
180:               {\setcounter{idxcols}{3}}
181:               % In this error case use default value and
182:               % produce a warning to inform the user.
183:               {\setcounter{idxcols}{2}
184:                \PackageWarningNoLine{fomdes}
185:                   {PrintIndex: Unknown column option '#2'.
186:                    Use 2 columns instead}}}}
187:         % Setup multi-column (if requested).
188:         \if@useidxcols
189:           % Change column's gap (default is 1.5pc).
190:           \setlength{\columnsep}{20pt}
191:           \begin{multicols}{\value{idxcols}}
192:         \fi
193:         % Set a gap between every single index entry.
194:         \setlength{\parskip}{\z@ \@plus 0.5\p@}%
195:         % Use index item entry as "the item".
196:         \let\item\@idxitem}
197:         % End of 'theindex'.
198:         {\if@useidxcols\end{multicols}\fi}
199:       % Force a page break only if it's requested.
200:       \if@desusenewpage\clearpage\fi
201:     % Used to fix a bug within hyperef-package.
202:       \ifpdf\phantomsection\fi
203:     % Create the Index.
204:       \printindex%
205:       % Force a page break only if it's requested.
206:       \if@desusenewpage\clearpage\fi
207:     }%
208:   \else
209:     % The macro \PrintIndex will be also available if the
```

```
210:     % package option 'index' wasn't used. This avoids error
211:     % messages while compilation!
212:     \newcommand{\PrintIndex}[1][]{%
213:       \@ifnextchar[%
214:         {\@iiPrintIndex[#1]}
215:         {\@iiPrintIndex[#1][]}%]
216:     }
217:     \long\def\@iiPrintIndex[#1][#2]{}
218:   \fi
219:   %------------------------------------------------------------
220:   % Table of code listings' depending settings and definitions.
221:   %------------------------------------------------------------
222:   % A table of code listings should be used within the current
223:   % document. 'if@uselistings' is TRUE when the user loads the
224:   % package with option 'listings'. By default 'if@uselistings'
225:   % is FALSE.
226:   \if@uselistings
227:     % Definition of macro which creates the table of code
228:     % listings. This macro supports one optional parameter
229:     % which can be used to fine tune the table of code
230:     % listings' layout.
231:     % Param1: Represents the table of code listings' heading
232:     %         to be used. The default heading is set to
233:     %         'Listings'.
234:     \newcommand{\PrintListings}[1][Listings]{%
235:     % Do not create a table of code listings if the listings
236:     % original heading macro does not exist! Possibly the
237:     % package 'fomfod' was not loaded beforehand!
238:       \ifx\lstlistlistingname\@undefined\else
239:         % Set listings' heading (defualt is 'Listings'),
240:         \renewcommand{\lstlistlistingname}{#1}
241:         % Force a page break only if it's requested.
242:         \if@desusenewpage\clearpage\fi
243:         % Used to fix a bug within hyperef-package.
244:         \ifpdf\phantomsection\fi
245:         % The table of code listings always belongs to the TOC!
246:         \addcontentsline{toc}{section}{\lstlistlistingname}
247:         % Create the table of code listings.
248:         \lstlistoflistings%
249:         % Force a page break only if it's requested.
250:         \if@desusenewpage\clearpage\fi
251:       \fi
252:     }%
253:   \else
254:     % The macro \PrintListings will be also available if the
255:     % package option 'listings' wasn't used. This avoids error
256:     % messages while compilation!
257:     \newcommand{\PrintListings}[1][]{}
258:   \fi
259:   %------------------------------------------------------------
260:   % Glossary depending settings and definitions.
```

```
261:  %---------------------------------------------------------
262:  % The glossary should be used within the current document.
263:  % 'if@useglossary' is TRUE when the user loads the package
264:  % with option 'glossary'. By default 'if@useglossary' is
265:  % FALSE.
266:  \if@useglossary
267:    % As mentioned in the package documentation the usage of
268:    % the glossary produces a message to inform the user about
269:    % "How to handle the glossary creation". Just print this
270:    % user message.
271:    \typeout{%
272:      *************************************************^^J%
273:      * PLEASE NOTE! The usage of a glossary requires to^^J%
274:      * run: bibtex <file>.gls^^J%
275:      *************************************************}%
276:    % Load package 'gloss' with option 'german' to support
277:    % german glossary look and feel and if reference page
278:    % is enabled the page of its first occurrence will also
279:    % be displayed.
280:    \if@useglossrefpage
281:      \RequirePackage[german,refpages]{gloss}
282:      \renewcommand{\glosspage}[1]{\space(Seite~#1)}
283:    \else
284:      \RequirePackage[german]{gloss}
285:    \fi
286:    % Ensures non-grouping mode and no indention. Otherwise
287:    % every glossary group (alphabetic order) will be led by
288:    % a group letter and every glossary entry has an indention.
289:    \renewcommand{\glossheading}[1]{}
290:    \renewcommand{\glosshang}{0pt}
291:    % The glossary has to be activated.
292:    \makegloss
293:    % Definition of macro which creates the glossary. This
294:    % macro defines one obligational parameter and supports
295:    % two optional parameters. All these parameters can be
296:    % used to fine tune the layout of glossary.
297:    % Param1: Represents the glossary's heading to be used.
298:    %         The default heading is set to 'Glossar'.
299:    % Param2: Can be used to take over all available entries
300:    %         from the glossary's database. The following
301:    %         pre-defined key words are available to this
302:    %         parameter:
303:    %         all: All entries from the glossary's database
304:    %              will be available within the document.
305:    %              The default is "Only referenced entries
306:    %              will be used within glossary".
307:    %              Remark: The usage of this parameter will
308:    %                      cause the 'gloss' package to
309:    %                      produce a package warning which
310:    %                      can be ignored!
311:    % Param3: This obligational parameter defines the name of
```

```
312:     %         the file which contains the glossary's library
313:     %         to be used with the glossary.
314:     \newcommand{\PrintGlossary}[1][Glossar]{%
315:       \@ifnextchar[%
316:         {\@iiPrintGlossary[#1]}
317:         {\@iiPrintGlossary[#1][]}%]
318:     }
319:     \long\def\@iiPrintGlossary[#1][#2]#3{%
320:       % Set glossary's heading (defualt is 'Glossar').
321:       \renewcommand{\glossname}{#1}
322:       % Force a page break only if it's requested.
323:       \if@desusenewpage\clearpage\fi
324:       % Used to fix a bug within hyperef-package.
325:       \ifpdf\phantomsection\fi
326:       % Use all entries from glossary database if requested.
327:       \ifthenelse{\equal{#2}{all}}{\gloss[nocite]{*}}{}
328:       % Create the glossary based on given database.
329:       \printgloss{#3}
330:       % Force a page break only if it's requested.
331:       \if@desusenewpage\clearpage\fi
332:     }%
333:   \else
334:     % The macro \PrintGlossary will be also available if the
335:     % package option 'glossary' wasn't used. This avoids error
336:     % messages while compilation!
337:     \newcommand{\PrintGlossary}[1][]{%
338:       \@ifnextchar[%
339:         {\@iiPrintGlossary[#1]}
340:         {\@iiPrintGlossary[#1][]}%]
341:     }
342:     \long\def\@iiPrintGlossary[#1][#2]#3{}
343:   \fi
344:   %-----------------------------------------------------------
345:   % Bibliography depending settings and definitions.
346:   %-----------------------------------------------------------
347:   % The bibliography should be used within the current
348:   % document. 'if@usebibliography' is TRUE by default. When
349:   % the user loads the package with option 'nobibliography'
350:   % 'if@usebibliography' is set to FALSE.
351:   \if@usebibliography
352:     % As mentioned in the package documentation the usage of
353:     % the bibliography produces a message to inform the user
354:     % about "How to handle the bibliography creation". Just
355:     % print this user message.
356:     \typeout{%
357:       ***************************************************^^J%
358:       * PLEASE NOTE! The usage of a bibliography requires^^J%
359:       * to run: bibtex <file>.aux^^J%
360:       ***************************************************}%
361:     % Load package 'natbib' with options 'square' and 'numbers'
362:     % to support DIN equivalent look and feel of bibliography,
```

```
363:     % which shows references as '[<number>]'.
364:     \RequirePackage[square,numbers]{natbib}
365:     % Definition of macro which creates the bibliography. This
366:     % macro defines one obligational parameter and supports
367:     % two optional parameters. All these parameters can be
368:     % used to fine tune the layout of bibliography.
369:     % Param1: Represents the bibliography's heading to be used.
370:     %         The default heading is set to 'Literatur'.
371:     % Param2: Represents the style of the bibliography. It
372:     %         sets the name of a file which contains the
373:     %         concrete style definition. As default value the
374:     %         style file 'dinat.bst' (without its extension)
375:     %         is used.
376:     % Param3: This obligational parameter defines the name
377:     %         of the file which contains the bibliography's
378:     %         library to be used with the bibliography.
379:     \newcommand{\PrintBibliography}[1][Literatur]{%
380:       \@ifnextchar[%
381:         {\@iiPrintBibliography[#1]}
382:         {\@iiPrintBibliography[#1][dinat]}%]
383:     }
384:     \long\def\@iiPrintBibliography[#1][#2]#3{%
385:       % Set bibliography's heading (defualt is 'Literatur').
386:       \renewcommand{\refname}{#1}
387:       % Force a page break only if it's requested.
388:       \if@desusenewpage\clearpage\fi
389:       % Used to fix a bug within hyperef-package.
390:       \ifpdf\phantomsection\fi
391:       % The bibliography always belongs to the TOC!
392:       \addcontentsline{toc}{section}{\refname}
393:       % Use given bibliography style.
394:       \bibliographystyle{#2}
395:       % Create the bibliography based on given database.
396:       \bibliography{#3}
397:       % Force a page break only if it's requested.
398:       \if@desusenewpage\clearpage\fi
399:     }%
400:   \else
401:     % The macro \PrintBibliography will be also available if
402:     % the package option 'glossary' wasn't used. This avoids
403:     % error messages while compilation!
404:     \newcommand{\PrintBibliography}[1][]{%
405:       \@ifnextchar[%
406:         {\@iiPrintBibliography[#1]}
407:         {\@iiPrintBibliography[#1][]}%]
408:     }
409:     \long\def\@iiPrintBibliography[#1][#2]#3{}
410:   \fi
411:   %-----------------------------------------------------------
412:   % Further macro definitions.
413:   %-----------------------------------------------------------
```

```
414:  % Defines a new style for the "see also" command with an
415:  % arrow followed by a slanted text.
416:  \newcommand{\SeeAlso}[1]{~$\to$~\textsl{#1}}
417:  %
418:  %--------------------------------------------------------
419:  % Finish style file.
420:  %%-------------------------------------------------------
```

A.4 Absatzdefinition (Quelldatei `fompar.sty`)

```
1:   %%-------------------------------------------------------
2:   %
3:   % File:        fompar.sty (FOM Paragraph Definition)
4:   %
5:   % Description: Definition of specific paragraph formatting
6:   %              macros for a usage in conjunction with "FOM
7:   %              Scientific Document Template".
8:   %
9:   % Author:      Axel Kesseler
10:  %
11:  % Version:     1.0.7
12:  %
13:  % Date:        2007/10/29
14:  %
15:  %--------------------------------------------------------
16:  \NeedsTeXFormat{LaTeX2e}[1995/12/01]
17:  \ProvidesPackage{fompar}
18:    [2007/10/29 v1.0 FOM Paragraph Definition]
19:  %--------------------------------------------------------
20:  % Declaration of all package options.
21:  %--------------------------------------------------------
22:  % --- No options defined at the moment! ---
23:  % Print a warning if the package receives an unknown option
24:  % just to inform the user.
25:  \DeclareOption*{\PackageWarningNoLine{fompar}
26:    {Unknown option '\CurrentOption'}}
27:  \ProcessOptions\relax%
28:  %--------------------------------------------------------
29:  % Load all depending packages.
30:  %--------------------------------------------------------
31:  % Load package 'ulem' with option 'normalem' to support
32:  % single underlined text.
33:  \RequirePackage[normalem]{ulem}
34:  %--------------------------------------------------------
35:  % Some configuration settings at the begin of a document.
36:  %--------------------------------------------------------
37:  % All pre-configured paragraph settings should take place
38:  % at the beginnig of a document.
39:  \AtBeginDocument{%
40:    \DefaultRemarkIndent
```

```
41:     \DefaultRemarkOffset
42:     \DefaultItemIndent
43:     \DefaultItemOffset
44:   }%
45:   %-----------------------------------------------------------
46:   % A paragraph with label (left aligned) and body.
47:   %-----------------------------------------------------------
48:   % A length value to store left indention globally. The value
49:   % of 'remark@leftmargin' will be used by macro \Remark.
50:   \newlength{\remark@leftmargin}
51:   % This macro is used to change left indention of paragraphs
52:   % globally. It resets the value of 'remark@leftmargin' which
53:   % is used by macro \Remark. It defines one obligational
54:   % parameters to setup left indention.
55:   % Param1: This obligational parameter set the left indention
56:   %         of paragraphs created by macro \Remark.
57:   \newcommand{\RemarkIndent}[1]{%
58:     \setlength{\remark@leftmargin}{#1}}
59:   % This macro is used the reset left indention of paragraphs
60:   % created by macro \Remark to its default indention of 3cm.
61:   % This macro does not have parameters.
62:   \newcommand{\DefaultRemarkIndent}{%
63:     \RemarkIndent{3cm}}
64:   % A length value to store paragraph's leading gap globally.
65:   % The value of 'remark@topsep' will be used by macro \Remark.
66:   \newlength{\remark@topsep}
67:   % This macro is used to change leading gap of paragraphs
68:   % globally. It resets the value of 'remark@topsep' which
69:   % is used by macro \Remark. It defines one obligational
70:   % parameters to setup leading gap.
71:   % Param1: This obligational parameter set the leading gap
72:   %         of paragraphs created by macro \Remark.
73:   \newcommand{\RemarkOffset}[1]{%
74:     \setlength{\remark@topsep}{#1}}
75:   % This macro is used the reset leading gap of paragraphs
76:   % created by macro \Remark to its default value of \parskip.
77:   % This macro does not have parameters.
78:   \newcommand{\DefaultRemarkOffset}{%
79:     \RemarkOffset{\parskip}}
80:   % Definition of macro which supports a paragraph formatting
81:   % with a label and a body like a desription. The label of
82:   % the paragraph will be LEFT aligned (the only difference
83:   % to macro \Item). This macro defines two obligational
84:   % parameters and two optional parameters.
85:   % Param1: Represents the paragraph's left indention. The
86:   %         label will have 90% of this width and the gap
87:   %         between label and body will be set to the rest.
88:   % Param2: Represents the paragraph's gap to the text
89:   %         before.
90:   % Param3: This obligational parameter is used as paragraph's
91:   %         label and can be formatted by the user.
```

```
 92:  % Param4: This obligational parameter is used as paragraph's
 93:  %         body and can contain multiple paragraphs.
 94:  % Sample: The recommended usage of this macro is for
 95:  %         paragraphs which explain something special like:
 96:  %         Attention:   This is not allowed for
 97:  %                      an administrator!
 98:  \newcommand{\Remark}[1][\remark@leftmargin]{%
 99:    \@ifnextchar[%
100:      {\@iiRemark[#1]}
101:      {\@iiRemark[#1][\remark@topsep]}%]
102:  }%
103:  \long\def\@iiRemark[#1][#2]#3#4{%
104:    \begin{list}{}{%
105:      \setlength{\topsep}{#2}%
106:      \setlength{\parsep}{\parskip}%
107:      \setlength{\leftmargin}{#1}%
108:      \setlength{\labelwidth}{0.9\leftmargin}%
109:      \setlength{\labelsep}{0.1\leftmargin}%
110:      \renewcommand{\makelabel}[1]{##1\ \hfil}}%
111:    \item[#3]#4%
112:    \end{list}%
113:  }%
114:  %-----------------------------------------------------------
115:  % A paragraph with a label (right aligned) and a body.
116:  %-----------------------------------------------------------
117:  % A length value to store left indention globally. The value
118:  % of 'item@leftmargin' will be used by macro \Item.
119:  \newlength{\item@leftmargin}
120:  % This macro is used to change left indention of paragraphs
121:  % globally. It resets the value of 'item@leftmargin' which
122:  % is used by macro \Item. It defines one obligational
123:  % parameters to setup left indention.
124:  % Param1: This obligational parameter set the left indention
125:  %         of paragraphs created by macro \Item.
126:  \newcommand{\ItemIndent}[1]{%
127:    \setlength{\item@leftmargin}{#1}}
128:  % This macro is used the reset left indention of paragraphs
129:  % created by macro \Item to its default indention of 4cm.
130:  % This macro does not have parameters.
131:  \newcommand{\DefaultItemIndent}{%
132:    \ItemIndent{4cm}}
133:  % A length value to store paragraph's leading gap globally.
134:  % The value of 'item@topsep' will be used by macro \Item.
135:  \newlength{\item@topsep}
136:  % This macro is used to change leading gap of paragraphs
137:  % globally. It resets the value of 'item@topsep' which
138:  % is used by macro \Item. It defines one obligational
139:  % parameters to setup leading gap.
140:  % Param1: This obligational parameter set the leading gap
141:  %         of paragraphs created by macro \Item.
142:  \newcommand{\ItemOffset}[1]{%
```

```
143:     \setlength{\item@topsep}{#1}}
144:   % This macro is used the reset leading gap of paragraphs
145:   % created by macro \Item to its default value of \parskip.
146:   % This macro does not have parameters.
147:   \newcommand{\DefaultItemOffset}{%
148:     \ItemOffset{\parskip}}
149:   % Definition of macro which supports a paragraph formatting
150:   % with a label and a body like a desription. The label of
151:   % the paragraph will be RIGHT aligned (the only difference
152:   % to macro \Remark). This macro defines two obligational
153:   % parameters and two optional parameters.
154:   % Param1: Represents the paragraph's left indention. The
155:   %         label will have 90% of this width and the gap
156:   %         between label and body will be set to the rest.
157:   % Param2: Represents the paragraph's gap to the text
158:   %         before.
159:   % Param3: This obligational parameter is used as paragraph's
160:   %         label and can be formatted by the user.
161:   % Param4: This obligational parameter is used as paragraph's
162:   %         body and can contain multiple paragraphs.
163:   % Sample: The recommended usage of this macro is for
164:   %         paragraphs which explain for example source
165:   %         code commands like:
166:   %              rm -rf: This command shouldn't be used
167:   %                      within the root directory!
168:   \newcommand{\Item}[1][\item@leftmargin]{%
169:     \@ifnextchar[%
170:       {\@iiItem[#1]}
171:       {\@iiItem[#1][\item@topsep]}%]
172:   }%
173:   \long\def\@iiItem[#1][#2]#3#4{%
174:     \begin{list}{}{%
175:       \setlength{\topsep}{#2}%
176:       \setlength{\parsep}{\parskip}%
177:       \setlength{\leftmargin}{#1}%
178:       \setlength{\labelwidth}{0.9\leftmargin}%
179:       \setlength{\labelsep}{0.1\leftmargin}}%
180:     \item[#3]#4%
181:     \end{list}%
182:   }%
183:   %-----------------------------------------------------------
184:   % Macros for special paragraph representation.
185:   %-----------------------------------------------------------
186:   % Definition of macro which supports a paragraph with a kind
187:   % of headline. The difference to the LeTeX-owned macros
188:   % \paragraph and \subparagraph is, that the heading of macro
189:   % \Description will NEVER be a part of TOC! Furthermore the
190:   % heading uses a separate text line. This macro defines two
191:   % obligational parameters and one optional parameter.
192:   % Param1: Represents the paragraph's heading layout. The
193:   %         following pre-defined key words are available to
```

```
194:  %         this parameter:
195:  %         n: The heading given with second parameter is
196:  %            not formatted.
197:  %         b: The heading given with second parameter is
198:  %            bold (=default).
199:  %         s: The heading given with second parameter is
200:  %            slanted.
201:  %         u: The heading given with second parameter is
202:  %            underlined.
203:  % Param2: This obligational parameter is used as paragraph's
204:  %         heading.
205:  % Param3: This obligational parameter is used as paragraph's
206:  %         body.
207:  \newcommand{\Description}[3][b]{%
208:    \vspace{\parskip}% Leading space
209:    % Headline formatting requested through parameter #1.
210:    \if      #1n{#2}%
211:    \else\if #1s{\textsl{#2}}%
212:    \else\if #1u{\uline {#2}}%
213:    \else       {\textbf{#2}}%
214:    \fi\fi\fi%
215:    \nopagebreak\par\vspace{-.5\parskip}{#3}%
216:  }%
217:  %-----------------------------------------------------------
218:  % A paragraph with left and right indention.
219:  %-----------------------------------------------------------
220:  % Definition of an environment which allows a left and
221:  % right paragraph indention. Both indentions can be set
222:  % independently. This macro defines two obligational
223:  % parameters and one optional parameter.
224:  % Param1: Represents the paragraph's leading gap. The value
225:  %         of \parskip is used by default.
226:  % Param2: This obligational parameter is used to set
227:  %         paragraph's left indention.
228:  % Param3: This obligational parameter is used to set
229:  %         paragraph's right indention.
230:  \newenvironment{Citation}[3][\parskip]{%
231:    \begin{list}{}{%
232:      \setlength{\partopsep}{#1}%
233:      \setlength{\topsep}{0pt}%
234:      \setlength{\parsep}{\parskip}%
235:      \setlength{\itemsep}{0pt}%
236:      \setlength{\listparindent}{0pt}%
237:      \setlength{\labelwidth}{0pt}%
238:      \setlength{\labelsep}{0pt}%
239:      \setlength{\leftmargin}{#2}%
240:      \setlength{\rightmargin}{#3}%
241:    }\item[]}%
242:    {\end{list}%
243:  }%
244:  %
```

```
245:  %----------------------------------------------------------
246:  % Finish style file.
247:  %%---------------------------------------------------------
```

A.5 Gleitobjektdefinition (Quelldatei fomfod.sty)

```
 1:  %%---------------------------------------------------------
 2:  %
 3:  % File:        fomfod.sty (FOM Floating Object Definition)
 4:  %
 5:  % Description: Definition of specific floating object
 6:  %              formatting macros for a usage in conjunction
 7:  %              with "FOM Scientific Document Template".
 8:  %
 9:  % Author:      Axel Kesseler
10:  %
11:  % Version:     1.0.10
12:  %
13:  % Date:        2007/10/29
14:  %
15:  %----------------------------------------------------------
16:  \NeedsTeXFormat{LaTeX2e}[1995/12/01]
17:  \ProvidesPackage{fomfod}
18:    [2007/10/29 v1.0 FOM Floating Object Definition]
19:  %----------------------------------------------------------
20:  % Declaration of all package options.
21:  %----------------------------------------------------------
22:  % Option 'classiccaption' enables the usage of captions
23:  % with "classic text layout". This means the usage of a
24:  % font with "serifs". A sans serif font is used by default.
25:  \newif\if@usestdcaption
26:  \@usestdcaptionfalse
27:  \DeclareOption{classiccaption}
28:    {\@usestdcaptiontrue}
29:  % Option 'listingcolored' enables the usage of colored
30:  % source code listings. Source code listings are uncolored
31:  % (black/white) by default.
32:  \newif\if@uselstcolor
33:  \@uselstcolorfalse
34:  \DeclareOption{listingcolored}
35:    {\@uselstcolortrue}
36:  % Print a warning if the package receives an unknown option
37:  % just to inform the user.
38:  \DeclareOption*{\PackageWarningNoLine{fomfod}
39:    {Unknown option '\CurrentOption'}}
40:  \ProcessOptions\relax%
41:  %----------------------------------------------------------
42:  % Avoid package loading when 'letter' class is loaded!
43:  %----------------------------------------------------------
44:  \@ifclassloaded{letter}{%
```

```
45:     \PackageError{fomfod}{^^JThis package cannot be used with %
46:                              document class 'letter'!^^J}
47:     \endinput%
48:   }{}% Build break!
49:   %-----------------------------------------------------------
50:   % Bugfix for the 'beamer' class!
51:   %-----------------------------------------------------------
52:   %% BUGFIX: Usage of class 'beamer' produces at lot of errors!
53:   %%         To avoid these error the usage of package 'subfig'
54:   %%         will be disabled. Furthermore, the package 'caption'
55:   %%         needs to define the macro '\@makecaption' while
56:   %%         using this package together with class 'beamer'!
57:   \newif\if@usebeamer
58:   \@ifclassloaded{beamer}{%
59:     \@usebeamertrue%
60:   }{%
61:     \@usebeamerfalse%
62:   }%
63:   %-----------------------------------------------------------
64:   % Load all depending packages.
65:   %-----------------------------------------------------------
66:   %% Fix bugs in 'beamer' class before loading package 'caption'!
67:   \if@usebeamer% Bugfix for 'caption' and 'beamer' packages!
68:     \@ifundefined{@makecaption}{%
69:       \PackageWarning{fomfod}{^^JThe beamer class is used! %
70:             In this case caption^^Jpackage macro 'makecaption' %
71:             will be pre-defined!^^J}
72:       \newcommand\@makecaption[2]{}%
73:   }\fi% End of: Bugfix for 'caption' package!
74:   % Used to support different caption layouts.
75:   \RequirePackage{caption}
76:   % Used to support graphics/figures within a document.
77:   \RequirePackage{graphicx}
78:   \if@usebeamer%
79:     \PackageWarning{fomfod}{^^JThe beamer class is used! In %
80:           this case^^Jpackage subfig is not supported!^^J}
81:   \else% Disable 'subfig' package only if 'beamer' class is loaded!
82:     % Used to support sub-figures within a figure.
83:     \RequirePackage{subfig}
84:   \fi% End of disabling 'subfig' package!
85:   % Internally used to check macro parameters.
86:   \RequirePackage{ifthen}
87:   % Used to support tables with a defined column width.
88:   \RequirePackage{tabularx}
89:   % Used to support source code listings within a document.
90:   \RequirePackage{listings}
91:   % Used to support different colors within a document.
92:   \RequirePackage{color}
93:   %-----------------------------------------------------------
94:   % Predefine caption depending styles.
95:   %-----------------------------------------------------------
```

```
 96:  % Global setup for all supported captions depending on
 97:  % package option 'classiccaption'.
 98:  \if@usestdcaption
 99:    % Define standard caption styles.
100:    \DeclareCaptionFont{capfont}{\small}
101:    \DeclareCaptionFont{lblfont}{\bfseries}
102:    \DeclareCaptionFont{txtfont}{}
103:    % Also adapt text style for source boxes.
104:    \def\@srcboxfmt#1{\tiny{#1}}
105:  \else
106:    % Define modern caption styles.
107:    \DeclareCaptionFont{capfont}{\small}
108:    \DeclareCaptionFont{lblfont}{\bfseries\sffamily}
109:    \DeclareCaptionFont{txtfont}{\sffamily}
110:    % Also adapt text style for source boxes.
111:    \def\@srcboxfmt#1{\tiny{\textsf{#1}}}
112:  \fi
113:  % Always use this type of a separator!
114:  \DeclareCaptionLabelSeparator{lblsep}{:\quad}
115:  %-----------------------------------------------------------
116:  % Adjust all supported captions.
117:  %-----------------------------------------------------------
118:  % Adjust figure's caption.
119:  \captionsetup[figure]{format=hang}
120:  \captionsetup[figure]{format=hang}
121:  \captionsetup[figure]{font=capfont}
122:  \captionsetup[figure]{textfont=txtfont}
123:  \captionsetup[figure]{labelfont=lblfont}
124:  \captionsetup[figure]{labelsep=lblsep}
125:  % Adjust sub-figure's caption.
126:  \captionsetup[subfigure]{margin=0pt}
127:  \captionsetup[subfigure]{font=capfont}
128:  \captionsetup[subfigure]{listofformat=subparens}
129:  \captionsetup[subfigure]{labelformat=parens}
130:  \captionsetup[subfigure]{labelsep=quad}
131:  % Adjusting table's caption.
132:  \captionsetup[table]{format=hang}
133:  \captionsetup[table]{margin=10pt}
134:  \captionsetup[table]{font=capfont}
135:  \captionsetup[table]{textfont=txtfont}
136:  \captionsetup[table]{labelfont=lblfont}
137:  \captionsetup[table]{labelsep=lblsep}
138:  \captionsetup[table]{aboveskip=0pt}
139:  % Adjusting listing's caption.
140:  \captionsetup[lstlisting]{format=hang}
141:  \captionsetup[lstlisting]{margin=10pt}
142:  \captionsetup[lstlisting]{font=capfont}
143:  \captionsetup[lstlisting]{textfont=txtfont}
144:  \captionsetup[lstlisting]{labelfont=lblfont}
145:  \captionsetup[lstlisting]{labelsep=lblsep}
146:  %-----------------------------------------------------------
```

```
147:  % Define a box for usage as source reference.
148:  %----------------------------------------------------------
149:  % This internally used macro create a box to place a floating
150:  % object's source reference. It defines two obligational
151:  % parameters and supports one optional parameter.
152:  % Param1: Represents the text alignment to be used. All
153:  %         alignment values of macro \makebox are available
154:  %         and right alignment will be used by default.
155:  % Param2: Represents source box's width. The value depends
156:  %         on the caller of this macro.
157:  % Param3: Represents source box's text to be printed out.
158:  %         The value depends on the caller of this macro.
159:  \newcommand{\@sourcebox}[3][r]{%
160:    \parbox[b][3mm]{#2}{%
161:      \makebox[\linewidth][#1]{\@srcboxfmt{#3}}}}
162:  %----------------------------------------------------------
163:  % General plain image formatting definition (plain figure).
164:  %----------------------------------------------------------
165:  % Definition of an environment which handles the usage of a
166:  % plain figure. This environment defines three obligational
167:  % parameters and supports two optional parameters. All these
168:  % parameters can be used to fine tune the layout of the figure.
169:  % Param1: Represents the percentaged width of the figure
170:  %         to be used depending on current text width. The
171:  %         default is set to 90% of text width.
172:  % Param2: Represents the usage of a box around the figure's
173:  %         content. The following pre-defined key word is
174:  %         available to this parameter:
175:  %         box: A box will be shown. The default is "do not
176:  %              show a box".
177:  % Param3: This obligational parameter defines the figure's
178:  %         caption to be used.
179:  % Param4: This obligational parameter defines the figure's
180:  %         reference label to be used.
181:  % Param5: This obligational parameter defines the figure's
182:  %         source description.
183:  % Sample: The following piece of code includes a self defined
184:  %         figure content and sets the caption, reference
185:  %         label and a source description (default width and
186:  %         no box is used):
187:  %         \begin{PlainFigure}{This is my first plain figure}
188:  %                            {fig:myfirstfigure}
189:  %                            {Source: www.ctan.org}
190:  %         Freely definable figure content...
191:  %         \end{PlainFigure}
192:  \newenvironment{PlainFigure}[1][0.9]{%
193:    \@ifnextchar[%
194:      {\begin{@iiPlainFigure}[#1]}
195:      {\begin{@iiPlainFigure}[#1][]}%]
196:  }{\end{@iiPlainFigure}}%
197:  % Needed to display a framed box around the minipage (has to
```

```
198:  % be defined outside any other macro).
199:  \newsavebox{\pf@box}
200:  % Definition of plain figure's environment begin section.
201:  \def\@iiPlainFigure[#1][#2]#3#4#5{%
202:    % Save given parameters for a later usage.
203:    \def\pf@width{#1\textwidth}
204:    \def\pf@caption{#3}
205:    \def\pf@label{#4}
206:    \def\pf@source{#5}
207:    % Prepare content settings depending on given parameter.
208:    \ifthenelse{\equal{#2}{box}}{%
209:      \def\pf@content{%
210:        \begin{lrbox}{\pf@box}
211:        \begin{minipage}{\pf@width}}
212:      \def\endpf@content{%
213:        \end{minipage}
214:        \end{lrbox}
215:        \setlength{\fboxsep}{5pt}
216:        \fbox{\usebox{\pf@box}}}}{%
217:      \def\pf@content{%
218:        \begin{minipage}{\pf@width}}
219:      \def\endpf@content{%
220:        \end{minipage}}}
221:    % Begin with environment start section.
222:    \begin{figure}[htb]%
223:    \bigskip%
224:    \centering%
225:    \begin{pf@content}
226:  }%
227:  % Definition of plain figure's environment end section.
228:  \def\end@iiPlainFigure{%
229:    % End with content environment.
230:    \end{pf@content}
231:    % Display source data and figure caption.
232:    \setlength{\abovecaptionskip}{4pt}
233:    \@sourcebox{\pf@width}{\pf@source}
234:    \parbox{\pf@width}{%
235:      \caption{\pf@caption}\protect\label{\pf@label}}
236:    % End all other environments.
237:    \end{figure}%
238:  }%
239:  %----------------------------------------------------------
240:  % Define some figure depending internal values.
241:  %----------------------------------------------------------
242:  % Definition of an if-statement to find out if a box should
243:  % be used around a figure. This 'if' has to be defined "out
244:  % of every block"! Otherwise an error message comes up while
245:  % compilation.
246:  \newif\if@usefigurebox
247:  % Define some internally used length values.
248:  \newlength{\@figurewidth}
```

```
249:  %-----------------------------------------------------------
250:  % General image formatting definition (single picture).
251:  %-----------------------------------------------------------
252:  % Definition of macro which handles the usage of a single
253:  % picture. This macro defines four obligational parameters
254:  % and supports two optional parameters. All these parameters
255:  % can be used to fine tune the layout of the figure.
256:  % Param1: Represents the percentaged width of the figure
257:  %         to be used depending on current text width. The
258:  %         default is set to 90% of text width.
259:  % Param2: Represents the usage of a box around the figure.
260:  %         The following pre-defined key word is available
261:  %         to this parameter:
262:  %         box: A box will be shown. The default is "do not
263:  %              show a box".
264:  % Param3: This obligational parameter defines the figure's
265:  %         caption to be used.
266:  % Param4: This obligational parameter defines the figure's
267:  %         reference label to be used.
268:  % Param5: This obligational parameter defines the figure's
269:  %         image file name without its extension to be used.
270:  % Param6: This obligational parameter defines the figure's
271:  %         source description.
272:  % Sample: The following piece of code includes an image file
273:  %         named 'myimage.jpg' and sets the caption, reference
274:  %         label and a source description (default width and
275:  %         no box is used):
276:  %         \SinglePicture{This is my first image}
277:  %                       {fig:myfirstimage}
278:  %                       {myimage}
279:  %                       {Source: www.ctan.org}
280:  \newcommand{\SinglePicture}[1][0.9]{%
281:      \@ifnextchar[%
282:        {\@iiSinglePicture[#1]}
283:        {\@iiSinglePicture[#1][]}%]
284:  }%
285:  \def\@iiSinglePicture[#1][#2]#3#4#5#6{%
286:    \ifthenelse{\equal{#2}{box}}
287:      {\@usefigureboxtrue}
288:      {\@usefigureboxfalse}
289:    \setlength{\@figurewidth}{#1\textwidth}
290:    \begin{figure}[htb]%
291:      \bigskip%
292:      \centering%
293:      \if@usefigurebox
294:        \fbox{\includegraphics[width=\@figurewidth]{#5}}
295:      \else
296:        \includegraphics[width=\@figurewidth]{#5}
297:      \fi
298:      \setlength{\abovecaptionskip}{4pt}
299:      \@sourcebox{\@figurewidth}{#6}
```

```
300:       \parbox{\@figurewidth}{%
301:         \caption{#3}\protect\label{#4}}%
302:     \end{figure}%
303:   }%
304:   %------------------------------------------------------------
305:   % General image formatting definition (double picture).
306:   %------------------------------------------------------------
307:   % General remark to the "double picture" implementation!
308:   % The support of double pictures has to be split into two
309:   % separat macros. The main reason for this is the "number
310:   % of parameters" limitaion because LaTeX provides only up
311:   % to nine macro parameters!
312:   %
313:   % This macro definition acts as a kind of a "framework"
314:   % for its sub-pictures and it defines four obligational
315:   % parameters and supports one optional parameter. All
316:   % these parameters can be used to fine tune the layout
317:   % of the figure. Width of both images will be adjusted
318:   % to 42% of current text width automatically.
319:   % Param1: Represents the usage of a box around the
320:   %         sub-figure. The following pre-defined key word
321:   %         is available to this parameter:
322:   %         box: A box will be shown. The default is "do not
323:   %              show a box".
324:   % Param2: This obligational parameter defines the figure's
325:   %         main caption to be used.
326:   % Param3: This obligational parameter defines the figure's
327:   %         main reference label to be used.
328:   % Param4: Takes the definition of left image (see also
329:   %         macro \DoubleSubPicture).
330:   % Param5: Takes the definition of right image (see also
331:   %         macro \DoubleSubPicture).
332:   % Sample: See sample shown in \DoubleSubPicture.
333:   \newcommand{\DoublePicture}[5][]{%
334:     \ifthenelse{\equal{#1}{box}}
335:       {\@usefigureboxtrue}
336:       {\@usefigureboxfalse}
337:     \setlength{\@figurewidth}{0.42\textwidth}
338:     \begin{figure}[htb]%
339:       \centering%
340:       {#4}\qquad{#5}
341:       \parbox{2\@figurewidth}{% two images!
342:         \caption{#2}\protect\label{#3}}%
343:     \end{figure}%
344:   }%
345:   % This macro is used the setup the left/right image of
346:   % a double picture. It defines only four obligational
347:   % parameters.
348:   % Param1: This obligational parameter defines the
349:   %         sub-figure's caption to be used.
350:   % Param2: This obligational parameter defines the
```

```
351:  %         sub-figure's reference label to be used.
352:  % Param3: This obligational parameter defines the
353:  %         sub-figure's image file name without its
354:  %         extension to be used.
355:  % Param4: This obligational parameter defines the
356:  %         sub-figure's source description.
357:  % Sample: The following piece of code shows how a combination
358:  %         of \DoublePicture and \DoubleSubPicture should be
359:  %         used (box is not used):
360:  %         \DoublePicture{double picture main caption}
361:  %                       {fig:mainref}
362:  %                       {\DoubleSubPicture{left sub-caption}
363:  %                                         {fig:leftref}
364:  %                                         {leftimage}
365:  %                                         {www.ctan.org}}
366:  %                       {\DoubleSubPicture{rigth sub-caption}
367:  %                                         {fig:rigthref}
368:  %                                         {rigthimage}
369:  %                                         {www.ctan.org}}
370:  \newcommand{\DoubleSubPicture}[4]{%
371:    \if@usebeamer% Disable usage of '\subfloat' if 'beamer' is loaded!
372:      \parbox{\@figurewidth}{%
373:        Package \textsf{\textbf{subfig}} is not supported %
374:        if \textsf{\textbf{beamer}} class is loaded!\par
375:        \setlength{\parskip}{0pt}
376:        \@sourcebox{\@figurewidth}{#4}
377:      }%
378:    \else% Use macro '\subfloat'!
379:      \subfloat[#1\protect\label{#2}]{%
380:        \parbox{\@figurewidth}{%
381:          \if@usefigurebox
382:            \fbox{%
383:              \includegraphics[width=\@figurewidth]{#3}}%
384:          \else
385:            \includegraphics[width=\@figurewidth]{#3}%
386:          \fi
387:          \setlength{\parskip}{0pt}
388:          \@sourcebox{\@figurewidth}{#4}
389:        }%
390:      }%
391:    \fi% End of disabling usage of macro '\subfloat'!
392:  }%
393:  %-----------------------------------------------------------
394:  % Definition of 'tabularx' column alignment.
395:  %-----------------------------------------------------------
396:  % Definition of 'tabularx' specific column alignment types
397:  % which are used as a short cut only!
398:  % Right aligned column.
399:  \newcolumntype{R}{>{\raggedleft\arraybackslash}X}
400:  % Centered aligned column.
401:  \newcolumntype{C}{>{\centering\arraybackslash}X}
```

```
402:    % Left aligned column
403:    \newcolumntype{L}{>{\raggedright\arraybackslash}X}
404:    % Block aligned column (default).
405:    \newcolumntype{B}{X}
406:    %-----------------------------------------------------------
407:    % Definition of a pre-configured table environment.
408:    %-----------------------------------------------------------
409:    % Create needed box to calculate the width of enclosed table.
410:    \newsavebox{\tab@box}
411:    % Create needed dimension to store calculate table width.
412:    \newdimen{\tab@width}
413:    % This environment acts as a kind of "framework" to handle
414:    % special table formatting issues. It defines three
415:    % obligational parameters to setup the environment.
416:    % Param1: This obligational parameter defines the table's
417:    %         caption to be used.
418:    % Param2: This obligational parameter defines the table's
419:    %         reference label to be used.
420:    % Param3: This obligational parameter defines the table's
421:    %         source description.
422:    \newenvironment{Table}[3]{%
423:      % Save given source data for later usage.
424:      \def\tab@source{#3}%
425:      \begin{table}[htb]%
426:        \medskip%
427:        \centering%
428:        \caption{#1}\label{#2}\medskip%
429:        \begin{lrbox}{\tab@box}}{%         start measure box
430:        \end{lrbox}%                        stop measure box
431:        \setbox0=\hbox{\usebox{\tab@box}}% calc box width
432:        \tab@width=\wd0%                    save box width
433:        \usebox{\tab@box}%                  display box data
434:        \par\@sourcebox{\tab@width}{\tab@source\hspace{1em}}%
435:      \end{table}%
436:    }%
437:    %-----------------------------------------------------------
438:    % Some length values depending on listing environment.
439:    %-----------------------------------------------------------
440:    % Definition of an internal length used as code listing's
441:    % indention. The default indention is set to 5pt.
442:    \newlength\@lstindent
443:    \setlength{\@lstindent}{5pt}
444:    % Definition of an internal length used as code listing's
445:    % border margin. The default indention is set to 6pt.
446:    \newlength\@lstborder
447:    \setlength{\@lstborder}{6pt}
448:    % Definition of an internal length used as code listing's
449:    % default value for optional parameters. The default value
450:    % is set to 7pt.
451:    \newlength\@lstdefmargin
452:    \setlength{\@lstdefmargin}{7pt}
```

```
453:  % Definition of an internal totally use margin for code
454:  % listings. The default value is set to 0pt and will be
455:  % calculated at runtime.
456:  \newlength\@lstmargin
457:  \setlength{\@lstmargin}{0pt}
458:  %---------------------------------------------------------
459:  % Definition of listing environment.
460:  %---------------------------------------------------------
461:  % Definition of an environment which supports code listings
462:  % directly written within the document. It defines two
463:  % obligational parameters and supports one optional parameter
464:  % to setup the code listing.
465:  % Param1: Represents the margin to be used around the code
466:  %         listing. The value of \@lstdefmargin (=7pt) will
467:  %         be used by default.
468:  % Param2: This obligational parameter defines the code
469:  %         listing's caption to be used.
470:  % Param3: This obligational parameter defines the code
471:  %         listing's reference label to be used.
472:  \lstnewenvironment{CodeListing}[3][\@lstdefmargin]{%
473:    \@lstsetup{#1}{#2}{#3}}{}
474:  %---------------------------------------------------------
475:  % Definition of listing command for file input.
476:  %---------------------------------------------------------
477:  % Definition of macro which supports code listings read from
478:  % an external file. It defines three obligational parameters
479:  % and supports one optional parameter to setup the code
480:  % listing.
481:  % Param1: Represents the margin to be used around the code
482:  %         listing. The value of \@lstdefmargin (=7pt) will
483:  %         be used by default.
484:  % Param2: This obligational parameter defines the code
485:  %         listing's caption to be used.
486:  % Param3: This obligational parameter defines the code
487:  %         listing's reference label to be used.
488:  % Param4: This obligational parameter defines the file name
489:  %         to be used as code listing's input file.
490:  \newcommand{\CodeListingFile}[4][\@lstdefmargin]{%
491:    \@lstsetup{#1}{#2}{#3}\lstinputlisting{#4}}
492:  %---------------------------------------------------------
493:  % Special listing setup depends on given parameter.
494:  %---------------------------------------------------------
495:  % Definition of an internally used macro which really does
496:  % the work of supporting code listings. It defines three
497:  % obligational parameters to setup the code listing.
498:  % Param1: This obligational parameter defines the additional
499:  %         border width around a code listing. The real border
500:  %         width is calculated from value of \@lstborder plus
501:  %         value of this parameter.
502:  % Param2: This obligational parameter defines the code
503:  %         listing's caption to be used.
```

```
504:   % Param3: This obligational parameter defines the code
505:   %         listing's reference label to be used.
506:   \newcommand{\@lstsetup}[3]{%
507:     % Setup additional margin!
508:     \setlength{\@lstmargin}{\@lstborder}
509:     % Add additional indent to the listing (if needed).
510:     \addtolength{\@lstmargin}{#1}
511:     % Setup listing's left and right indent!
512:     \lstset{xleftmargin=\@lstmargin}
513:     \lstset{xrightmargin=\@lstmargin}
514:     % Set listing's caption.
515:     \lstset{caption={#2}}
516:     % Set listing's reference label.
517:     \lstset{label=#3}
518:   }%
519:   %-----------------------------------------------------------
520:   % Listings font size setup macro.
521:   %-----------------------------------------------------------
522:   % Definition of macro which adjusts the code listing's font
523:   % size to be used. It defines one optional parameter.
524:   % Param1: Represents the font size to be used with code
525:   %         listings. The following pre-defined key words are
526:   %         available to this parameter:
527:   %         small:  A smaller font size will be used. This
528:   %                 parameter uses \scriptsize and allows
529:   %                 about 74 characters per code line.
530:   %         normal: The normal (=default) font size will be
531:   %                 used. This parameter uses \footnotesize
532:   %                 and allows about 60 characters per code
533:   %                 line.
534:   %         large:  A bigger font size will be used. This
535:   %                 parameter uses \small and allows about 55
536:   %                 characters per code line.
537:   \newcommand{\ListingFontSize}[1][normal]{%
538:     % Option [normal] allows about 60 characters.
539:     \ifthenelse{\equal{#1}{normal}}
540:       {\lstset{basicstyle=\footnotesize\ttfamily}}
541:       % Option [large] allows about 55 characters.
542:       {\ifthenelse{\equal{#1}{large}}
543:         {\lstset{basicstyle=\small\ttfamily}}
544:         % Option [small] allows about 74 characters.
545:         {\ifthenelse{\equal{#1}{small}}
546:           {\lstset{basicstyle=\scriptsize\ttfamily}}
547:           {\lstset{basicstyle=\footnotesize\ttfamily}}}}
548:   }%
549:   %-----------------------------------------------------------
550:   % General setup for all listings.
551:   %-----------------------------------------------------------
552:   % Definition of macro which restore all pre-configured code
553:   % listing settings because the user has the opportunity to
554:   % change all code listing setting through macro \lstset
```

```
555:    % offered by 'listings' package. To give an user a chance
556:    % to return to the settings pre-configured by this package
557:    % this macro can be called. This macro does not support
558:    % parameters.
559:    \newcommand{\DefaultListingConfig}{%
560:      % Use color depending on package option.
561:      \definecolor{bkgcol}{rgb}{1,1,1}
562:      \if@uselstcolor
563:        \definecolor{remcol}{rgb}{0,0.5,0}
564:        \definecolor{strcol}{rgb}{0.5,0.5,0.5}
565:        \definecolor{keycol}{rgb}{0,0,1}
566:      \else
567:        \definecolor{remcol}{rgb}{0,0,0}
568:        \definecolor{strcol}{rgb}{0,0,0}
569:        \definecolor{keycol}{rgb}{0,0,0}
570:      \fi
571:      % Set defualt code language.
572:      \lstset{language=[ISO]C++}
573:      % Caption on the top of the listing.
574:      \lstset{captionpos=t}
575:      % Skip above the caption.
576:      \lstset{aboveskip=1.5\bigskipamount}
577:      % Skip below the caption.
578:      \lstset{belowcaptionskip=\medskipamount}
579:      % Skip below the listing.
580:      \lstset{belowskip=\bigskipamount}
581:      % Expand 'tab' into two spaces.
582:      \lstset{tabsize=2}
583:      % Set background color to 'lightgray'.
584:      \lstset{backgroundcolor=\color{bkgcol}}
585:      % Font and style setup (Default = [normal]).
586:      \lstset{basicstyle=\footnotesize\ttfamily}
587:      \lstset{commentstyle=\ttfamily\color{remcol}}
588:      \lstset{stringstyle=\ttfamily\color{strcol}}
589:      \lstset{keywordstyle=\color{keycol}}
590:      % Some further self-explaining settings.
591:      \lstset{showstringspaces=false}
592:      \lstset{breaklines=true}
593:      \lstset{breakatwhitespace=true}
594:      \lstset{prebreak={\dots{}}}
595:      \lstset{postbreak={\dots{}}}
596:    }%
597:    %-----------------------------------------------------------
598:    % Default setup at document's begin!
599:    %-----------------------------------------------------------
600:    % All pre-configured code listing settings should take place
601:    % at the beginnig of a document.
602:    \AtBeginDocument{%
603:      \DefaultListingConfig%
604:    }%
605:    %
```

```
606:    %-----------------------------------------------------------
607:    % Finish style file.
608:    %%----------------------------------------------------------
```

A.6 Vermischtes (Quelldatei fomisc.sty)

```
1:     %%----------------------------------------------------------
2:     %
3:     % File:        fomisc.sty (FOM Miscellaneous Definition)
4:     %
5:     % Description: Definition file of miscellaneous macros for
6:     %              a usage in conjunction with "FOM Scientific
7:     %              Document Template".
8:     %
9:     % Author:      Axel Kesseler
10:    %
11:    % Version:     1.0.9
12:    %
13:    % Date:        2007/10/29
14:    %
15:    %-----------------------------------------------------------
16:    \NeedsTeXFormat{LaTeX2e}[1995/12/01]
17:    \ProvidesPackage{fomisc}
18:      [2007/10/29 v1.0 FOM Miscellaneous Macros]
19:    %-----------------------------------------------------------
20:    % Declaration of all package options.
21:    %-----------------------------------------------------------
22:    % --- No options defined at the moment! ---
23:    % Print a warning if the package receives an unknown option
24:    % just to inform the user.
25:    \DeclareOption*{\PackageWarningNoLine{fomisc}
26:      {Unknown option '\CurrentOption'}}
27:    \ProcessOptions\relax%
28:    %-----------------------------------------------------------
29:    % Load all depending packages.
30:    %-----------------------------------------------------------
31:    % Used to move and/or resize text fragments.
32:    \RequirePackage{textfit}
33:    % Internally used to check macro parameters.
34:    \RequirePackage{ifthen}
35:    % Needed to use macro '\scalebox'
36:    \RequirePackage{graphicx}
37:    %-----------------------------------------------------------
38:    % Macro for a high-positioned text.
39:    %-----------------------------------------------------------
40:    % Definition of macro which creates a text raised above
41:    % the baseline. The printed result looks much better than
42:    % the usage of "math" style like 'I$^2$C which set letters
43:    % italic. This macro defines one obligational parameter.
44:    % Param1: Represents the text to be raised above the
```

```
45:  %        baseline.
46:  % Sample: The following piece of code produces the text
47:  %        fragment "... I2C is bus system ...":
48:  %        "... I\Raise{2}C is bus system ..."
49:  \newcommand{\Raise}[1]{%
50:    % Position 85% of X-height above base line.
51:    \raisebox{0.85ex}{%
52:      % Shrink to 70% of text size.
53:      \scalebox{0.7}{#1}}%
54:  }%
55:  %------------------------------------------------------------
56:  % Definition of DIN/ISO styled date macro.
57:  %------------------------------------------------------------
58:  % Definition of macro which typesets the current date
59:  % (like macro \today does) in a short form. It supports
60:  % two optional parameters which can be used to adjust
61:  % the date layout.
62:  % Param1: Represents the kind of date formatting to be
63:  %        used. Its default value is set to 'din' and the
64:  %        following pre-defined key words are available to
65:  %        this parameter:
66:  %        din: [default] Layout of current date will be
67:  %             set to 'DD.MM.YYYY' according to DIN 5008.
68:  %        iso: Layout of current date will be typeset to
69:  %             'YYYY-MM-DD' according to ISO 8601:2004
70:  %             and EN 28601.
71:  % Param2: Represents the date delimiter to be used. The
72:  %        default delimiter depends on given base format
73:  %        announce through parameter #1. In case of
74:  %        parameter #1 equals 'din' a dot ('.') is used
75:  %        as this default delimiter. If 'iso' is used
76:  %        within parameter #1 than a dash ('-') will be
77:  %        used as default delimiter. The following
78:  %        pre-defined key words are available to this
79:  %        parameter:
80:  %        def: [default] Either a dor or a dash is used
81:  %             as default delimiter.
82:  %        The usage of 'def' as the default key word
83:  %        makes sense in case of the usage of NO delimiter!
84:  %        In this case it is possible to use command
85:  %        '\TodayShort[iso][]' which produces a date like
86:  %        'YYYYMMDD', in other words without a delimiter.
87:  \DeclareRobustCommand\TodayShort[1][din]{%
88:    \@ifnextchar[%
89:      {\@iiTodayShort[#1]}
90:      {\@iiTodayShort[#1][def]}%]
91:  }%
92:  \def\@iiTodayShort[#1][#2]{%
93:    % Select date layout depending
94:    % on parameter #1
95:    \ifthenelse{\equal{#1}{iso}}
```

```
 96:       % According to ISO 8601:2004 and EN 28601 a date should
 97:       % be formatted like: YYYY-MM-DD
 98:       {%
 99:         % Adjust date delimiter depending on parameter #2
100:         \ifthenelse{\equal{#2}{def}}
101:           {\def\@delim{"~}}
102:           {\def\@delim{#2}}
103:         \number\year\@delim%
104:         % Use leading '0' if current month is less than
105:         % 10th of year!
106:         {\ifnum\month<10{\number 0}\fi\number\month\@delim}%
107:         % Use leading '0' if current day is less than
108:         % 10th of month!
109:         {\ifnum\day<10{\number 0}\fi\number\day}%
110:       }%
111:       % According to DIN 5008 a date should be formatted
112:       % like: DD.MM.YYYY
113:       {%
114:         % Adjust date delimiter depending on parameter #2
115:         \ifthenelse{\equal{#2}{def}}
116:           {\def\@delim{.}}
117:           {\def\@delim{#2}}
118:         % Use leading '0' if current day is less than
119:         % 10th of month!
120:         {\ifnum\day<10{\number 0}\fi\number\day\@delim}%
121:         % Use leading '0' if current month is less than
122:         % 10th of year!
123:         {\ifnum\month<10{\number 0}\fi\number\month\@delim}%
124:         \number\year%
125:       }%
126:   }%
127:   %------------------------------------------------------------
128:   % Definition of various symbols.
129:   %------------------------------------------------------------
130:   % Definition of macros which are used as a short cut only!
131:   % All these macros do not have parameters.
132:   % \Registered: Serves as a short cut of the "registered"
133:   %              symbol like '®'.
134:   \DeclareRobustCommand\Registered{$^{\scalebox{0.7}[0.65]{\textregistered}}$}%
135:   % \Copyright:  Serves as a short cut of the "copyright"
136:   %              symbol like '©'.
137:   \DeclareRobustCommand\Copyright{\textcopyright}%
138:   % \Trademark:  Serves as a short cut of the "trademark"
139:   %              symbol like 'TM'.
140:   \DeclareRobustCommand\Trademark{$^{\scalebox{0.55}[0.5]{TM}}$}%
141:   %------------------------------------------------------------
142:   % The authors first name as a sample for "jumping" text.
143:   %------------------------------------------------------------
144:   % This macro definition is used to show how a "jumping"
145:   % text like \LaTeX can be defined. This macro does not have
146:   % parameters.
```

```
147:  \newcommand{\Axel}{A%   First letter is normal.
148:    \raisebox{0.4ex}{%    Second letter moved up,
149:      \hspace{-0.13em}{%  then moved left,
150:        \scaletoheight{0.95ex}{x}}}% lower cases and smaller.
151:    \raisebox{-0.15ex}{%  Third letter moved down,
152:      \hspace{-0.07em}{%  then moved left,
153:        \scaletoheight{1.1ex}{E}}}% upper cases and bigger.
154:    \raisebox{0.2ex}{%    Fourth letter moved up,
155:      \hspace{-0.025em}{% then moved left,
156:        \scaletoheight{1.4ex}{l}}}}% lower cases and bigger.
157:  %
158:  %----------------------------------------------------------
159:  % Finish style file.
160:  %%---------------------------------------------------------
```

A.7 Indexkonfiguration (Quelldatei fomidx.ist)

```
1:  %%---------------------------------------------------------
2:  %
3:  % File:        fomidx.ist (FOM Index Style Definition)
4:  %
5:  % Description: This file contains configurations to create
6:  %              a suitable index based on DIN index style.
7:  %
8:  % Author:      Axel Kesseler
9:  %
10: % Version:     1.0.7
11: %
12: % Date:        2007/10/29
13: %
14: %----------------------------------------------------------
15: % Based on DIN index style (supports german umlauts).
16: include "base/din.ist"
17: % Needed to use 'makeindex -g ...'.
18: quote '+'
19: % Index group configuration.
20: group_skip     "\n\n  \\indexspace%\n  \\indexspace%\n  \\indexspace%\n\n"
21: lethead_prefix "{\\sffamily\\bfseries\\large "
22: lethead_suffix " }\\nopagebreak%\n  \\indexspace\\nopagebreak%\n"
23: lethead_flag   1
24: heading_prefix "{\\sffamily\\bfseries\\large "
25: heading_suffix " }\\nopagebreak%\n  \\indexspace\\nopagebreak%\n"
26: headings_flag  1
27: % Special group heading configuration.
28: numhead_positive "Zahlen"
29: numhead_negative "zahlen"
30: symhead_positive "Symbole"
31: symhead_negative "symbole"
32: % Index item configuration.
33: delim_0 " \\space\\dotfill\\quad "
```

```
34:  delim_1 " \\space\\dotfill\\quad "
35:  delim_2 " \\space\\dotfill\\quad "
36:  % Index passes more than three pages.
37:  delim_r "~--~"
38:  % Index passes two pages.
39:  suffix_2p "\\,f."
40:  % Index passes three pages.
41:  suffix_3p "\\,ff."
42:  % Index item prefix configuration.
43:  item_1  "\n    \\subitem\\hspace{-19pt}--, "
44:  item_2  "\n      \\subsubitem\\hspace{-15pt}--, "
45:  item_01 "\n    \\subitem\\hspace{-19pt}--, "
46:  item_12 "\n      \\subsubitem\\hspace{-15pt}--, "
47:  item_x1 "\n    \\subitem\\hspace{-19pt}--, "
48:  item_x2 "\n      \\subsubitem\\hspace{-15pt}--, "
49:  %
50:  %------------------------------------------------------------
51:  % Finish index style file.
52:  %%-----------------------------------------------------------
```

A.8 Installationsanweisung (Quelldatei readme.txt)

```
1:   ,,,,,,,,,|,,,,,,,,,|,,,,,,,,,|,,,,,,,,,|,,,,,,,,,|,,,,,,,,,|
2:
3:            --- FOM Scientific Document Template ---
4:
5:   Allgemein
6:   ---------
7:   Die Anweisungen zur Installation des Pakets 'fomsdt' im
8:   LaTeX-System sind in der mitgelieferten Dokumentation
9:   enthalten!
10:
11:  Zur Distribution des Pakets 'fomsdt' gehören die Dateien:
12:
13:  'fomsdt.ins' (Installationsskript)
14:  'fomsdt.dtx' (Installationsdaten)
15:  'readme.txt' (Kurzanleitung)
16:
17:  Dokumentation
18:  -------------
19:  Die Erzeugung der Dokumentation erfolgt durch Eingabe
20:  einer Reihe von Befehlen. Hierfür ist als Erstes im
21:  Verzeichnis mit den geladenen Dateien 'fomsdt.ins' und
22:  'fomsdt.dtx' eine entsprechende Shell oder Konsole zu
23:  öffnen. Daran anschließend werden die folgenden Befehle
24:  der Reihe nach eingegeben und ausgeführt.
25:
26:  $ pdflatex fomsdt.dtx
27:  $ pdflatex fomsdt.dtx
28:  $ makeindex -s gind.ist -o fomsdt.ind fomsdt.idx
```

```
29:  $ pdflatex fomsdt.dtx
30:
31:  Eventuell ist der Befehl 'pdflatex fomsdt.dtx' ein
32:  weiteres mal auszuführen.
33:
34:  Nun sollte im aktuellen Verzeichnis die Dokumentation
35:  in Form der Datei 'fomsdt.pdf' vorhanden sein.
36:
37:
38:
39:  EOF
```

A.9 Benutzerdokumentation (Quelldatei `fomman.tex`)

```
 1:  %%-----------------------------------------------------------
 2:  %%
 3:  %%        --- FOM Scientific Document Template ---
 4:  %%
 5:  %%   This file contains the  whole user documentation for
 6:  %%  the 'fomsdt' package consisting of the following files.
 7:  %%
 8:  %% fomsdt.cls -- FOM Scientific Document Template (the class)
 9:  %% fomdss.sty -- FOM Document Start Section
10:  %% fomdes.sty -- FOM Document End Section
11:  %% fompar.sty -- FOM Paragraph Definition
12:  %% fomfod.sty -- FOM Floating Object Definition
13:  %% fomisc.sty -- FOM Miscellaneous Definition
14:  %% fomidx.ist -- FOM Index Style Definition
15:  %%
16:  %%-----------------------------------------------------------
17:  %% Document loading setting (needed for \filexxx).
18:  %%-----------------------------------------------------------
19:  \def\FileName{fomsdt.dtx}
20:  \def\FileDate{2007/10/29}
21:  \def\FileVersion{v1.0}
22:  \def\FileInfo{FOM Scientific Document Template}
23:  %%-----------------------------------------------------------
24:  %% Document loading.
25:  %%-----------------------------------------------------------
26:  \documentclass[a4paper,12pt]{ltxdoc}
27:  \usepackage{ngerman}
28:  \usepackage[latin1]{inputenc}
29:  \usepackage[T1]{fontenc}
30:  \usepackage{color}
31:  \usepackage{hyperref}
32:  \usepackage{ifpdf}
33:  \usepackage[normalem]{ulem}
34:  %%-----------------------------------------------------------
35:  %% Hyperlink setup.
36:  %%-----------------------------------------------------------
```

```
37:  \definecolor{linkcolor}{rgb}{0,0,0}
38:  \hypersetup{
39:    pdfauthor={Axel Kesseler},
40:    pdftitle={fomsdt : FOM Scientific Document Template},
41:    pdfborder=0 0 0,
42:    colorlinks=true,
43:    plainpages=false,
44:    bookmarksopen=true,
45:    bookmarksnumbered=true,
46:    linkcolor=linkcolor,
47:    citecolor=linkcolor,
48:    filecolor=linkcolor,
49:    urlcolor=linkcolor}
50:  %%------------------------------------------------------------
51:  %% Paragraph setup.
52:  %%------------------------------------------------------------
53:  \setlength{\parskip}{0.5\baselineskip plus 0.5ex minus 0.3ex}
54:  \setlength{\parindent}{0mm}
55:  \sloppy
56:  %%------------------------------------------------------------
57:  %% Page setup and page boundary.
58:  %%------------------------------------------------------------
59:  \clubpenalty=10000
60:  \widowpenalty=10000\displaywidowpenalty=10000
61:  \addtolength\oddsidemargin {40pt}
62:  \addtolength\evensidemargin{40pt}
63:  \addtolength{\textwidth}  {-40pt}
64:  %%------------------------------------------------------------
65:  %% 'dos.sty' setup.
66:  %%------------------------------------------------------------
67:  \OnlyDescription
68:  \PageIndex % use page number instead of code line index
69:  \setcounter{IndexColumns}{2}
70:  \IndexPrologue{\section*{Index}}
71:  \DoNotIndex{\newcommand,\newenvironment}
72:  %%------------------------------------------------------------
73:  %% This code was originally copied from file 'doc.sty'
74:  %% Some adoptions and extensions has been made!
75:  %%------------------------------------------------------------
76:  \def\File{Datei}
77:  \def\Files{Dateien}
78:  \def\Command{Befehl}
79:  \def\Commands{Befehle}
80:  \def\Option{Option}
81:  \def\Options{Optionen}
82:  \def\Macro{Makro}
83:  \def\Macros{Makros}
84:  \def\Env{Umgebung}
85:  \def\Envs{Umgebungen}
86:  \def\Depend{Abhängigkeit}
87:  \def\Depends{Abhängigkeiten}
```

```
 88:  \def\Problems{Bekannte Probleme}
 89:  \makeatletter
 90:  \def\CommandIndex#1{%
 91:    \MakePrivateLetters{\print@env{#1}}%
 92:    \special@indexother{#1}{\Command}{\Commands}}%
 93:  \def\FileIndex#1{%
 94:    \MakePrivateLetters{\print@env{#1}}%
 95:    \special@indexother{#1}{\File}{\Files}}%
 96:  \def\EnvIndex#1{%
 97:    \MakePrivateLetters{\print@env{#1}}%
 98:    \special@indexother{#1}{\Env}{\Envs}}%
 99:  \def\OptionIndex#1{%
100:    \MakePrivateLetters{\print@env{#1}}%
101:    \special@indexother{#1}{\Option}{\Options}}%
102:  \def\MacroIndex#1{%
103:    \MakePrivateLetters{\print@macro{#1}}%
104:    \special@indexmacro{#1}{\Macro}{\Macros}}%
105:  \def\DependsIndex#1{%
106:    \MakePrivateLetters{\print@package{#1}}%
107:    \special@depends{#1}{\Depend}{\Depends}}%
108:  \def\print@macro#1{\strut\MacroFont\string#1}
109:  \def\print@env#1{\strut\MacroFont#1}
110:  \def\print@package#1{\strut\textsf{#1}}
111:  \def\special@indexmacro#1#2#3{%
112:    \@bsphack%
113:      \special@index{%
114:        \expandafter\@gobble\string#1\actualchar%
115:        \string\verb\quotechar*\verbatimchar\string#1\verbatimchar{%
116:          \space\scriptsize\textsl{#2}}}%
117:      \special@index{#3>%
118:        \expandafter\@gobble\string#1\actualchar%
119:        \string\verb\quotechar*\verbatimchar\string#1\verbatimchar}%
120:      \@esphack}%
121:  \def\special@indexother#1#2#3{%
122:    \@bsphack%
123:      \special@index{%
124:        \expandafter\string#1\actualchar{%
125:          \protect\ttfamily\string#1}{%
126:            \space\scriptsize\textsl{#2}}}%
127:      \special@index{#3>%
128:        \expandafter\string#1\actualchar{%
129:          \protect\ttfamily\string#1}}%
130:      \@esphack}%
131:  \def\special@depends#1#2#3{%
132:    \@bsphack%
133:      \special@index{%
134:        \expandafter\string#1\actualchar{%
135:          \protect\sffamily\string#1}{%
136:            \space\scriptsize\textsl{#2}}}%
137:      \special@index{#3>%
138:        \expandafter\string#1\actualchar{%
```

```
139:           \protect\sffamily\string#1}}%
140:       \@esphack}%
141:   \def\DescFile{\pagebreak[2]\leavevmode\@bsphack%
142:       \begingroup\MakePrivateLetters\desc@file}%
143:   \def\desc@file#1{\endgroup%
144:     \marginpar{\raggedleft\print@env{#1}}%
145:     \special@indexother{#1}{\File}{\Files}\@esphack\ignorespaces}%
146:   \def\DescOption{\pagebreak[2]\leavevmode\@bsphack%
147:     \begingroup\MakePrivateLetters\desc@opt}%
148:   \def\desc@opt#1{\endgroup%
149:     \marginpar{\raggedleft\print@env{#1}}%
150:     \special@indexother{#1}{\Option}{\Options}\@esphack\ignorespaces}%
151:   \def\DescMacro{\pagebreak[2]\leavevmode\@bsphack%
152:     \begingroup\MakePrivateLetters\desc@macro}%
153:   \def\desc@macro#1{\endgroup%
154:     \marginpar{\raggedleft\print@macro{#1}}%
155:     \special@indexmacro{#1}{\Macro}{\Macros}\@esphack\ignorespaces}%
156:   \def\DescEnv{\pagebreak[2]\leavevmode\@bsphack%
157:     \begingroup\MakePrivateLetters\desc@env}%
158:   \def\desc@env#1{\endgroup%
159:     \marginpar{\raggedleft\print@env{#1}}%
160:     \special@indexother{#1}{\Env}{\Envs}\@esphack\ignorespaces}%
161:   \def\DescDepends{\pagebreak[2]\leavevmode\@bsphack%
162:       \begingroup\MakePrivateLetters\desc@depends}%
163:   \def\desc@depends#1{\endgroup%
164:     \marginpar{\raggedleft\print@package{#1}}%
165:     \special@depends{#1}{\Depend}{\Depends}\@esphack\ignorespaces}%
166:   \newcommand\DescOnly{\pagebreak[2]\leavevmode\@bsphack%
167:     \begingroup\MakePrivateLetters\desc@only}%
168:   \def\desc@only#1{\endgroup%
169:     \marginpar{\raggedleft{\small\bfseries#1}}%
170:     \index{#1}\@esphack\ignorespaces}%
171:   \newcommand\DescOnlyNoIdx{\pagebreak[2]\leavevmode\@bsphack%
172:     \begingroup\MakePrivateLetters\desc@onlyex}%
173:   \def\desc@onlyex#1{\endgroup%
174:     \marginpar{\raggedleft{\footnotesize\bfseries#1}}%
175:     \@esphack\ignorespaces}%
176:   \makeatother
177:
178:   \newcommand\OptExplain[3][10mm]{%
179:     \begin{list}{}{%
180:       \setlength{\topsep}{\parskip}%
181:       \setlength{\parsep}{\parskip}%
182:       \setlength{\leftmargin}{#1}%
183:       \setlength{\labelwidth}{0.9\leftmargin}%
184:       \setlength{\labelsep}{0.1\leftmargin}%
185:       \renewcommand{\makelabel}[1]{##1\ \hfil}}%
186:     \item[#2]#3%
187:     \end{list}}
188:
189:   \newcommand{\See}[1]{\space\textsl{\scriptsize $\to$~#1}}
```

```
190:
191:    %%-----------------------------------------------------------
192:    %% Local macro declaration.
193:    %%-----------------------------------------------------------
194:    \DeclareRobustCommand{\package}[1]{\textsf{#1}}
195:    \DeclareRobustCommand{\defval}[1]{\textbf{\textsl{#1}}}
196:    \DeclareRobustCommand{\bs}{$\backslash$}
197:    \DeclareRobustCommand{\holder}[1]{%
198:      \textsl{\small$\langle$#1$\rangle$}}
199:    \DeclareRobustCommand{\file}[1]{\texttt{\small#1}}
200:    \DeclareRobustCommand{\option}[1]{\texttt{\small#1}}
201:    \DeclareRobustCommand{\optionex}[1]{%
202:      \texttt{\small\lbrack#1\rbrack}}
203:    \DeclareRobustCommand{\secfile}[1]{\texttt{#1}}
204:
205:    \newcommand{\FOMSDT}{\emph{FOM Scientific Document Template}}
206:    \newcommand{\FOMDSS}{\emph{FOM Document Start Section}}
207:    \newcommand{\FOMDES}{\emph{FOM Document End Section}}
208:    \newcommand{\FOMPAR}{\emph{FOM Paragraph Definition}}
209:    \newcommand{\FOMFOD}{\emph{FOM Floating Object Definition}}
210:    \newcommand{\FOMISC}{\emph{FOM Miscellaneous Definition}}
211:    \newcommand{\FOMIDX}{\emph{FOM Index Style Definition}}
212:
213:    \newcommand{\fomsdt}{\package{fomsdt}}
214:    \newcommand{\fomdss}{\package{fomdss}}
215:    \newcommand{\fomdes}{\package{fomdes}}
216:    \newcommand{\fompar}{\package{fompar}}
217:    \newcommand{\fomfod}{\package{fomfod}}
218:    \newcommand{\fomisc}{\package{fomisc}}
219:    \newcommand{\fomidx}{\package{fomidx}}
220:
221:    \newcommand{\fomsdtpdf}{fomsdt.pdf}
222:    \newcommand{\fomsdtdtx}{fomsdt.dtx}
223:    \newcommand{\fomsdtins}{fomsdt.ins}
224:    \newcommand{\readmetxt}{readme.txt}
225:    \newcommand{\fomsdtcls}{fomsdt.cls}
226:    \newcommand{\fomdsssty}{fomdss.sty}
227:    \newcommand{\fomdessty}{fomdes.sty}
228:    \newcommand{\fomparsty}{fompar.sty}
229:    \newcommand{\fomfodsty}{fomfod.sty}
230:    \newcommand{\fomiscsty}{fomisc.sty}
231:    \newcommand{\fomidxist}{fomidx.ist}
232:
233:    \newenvironment{summary}{\small%
234:      \begin{center}{\bfseries \abstractname\vspace{-.5em}}\end{center}%
235:      \list{}{%
236:        \listparindent 0em%
237:        \itemindent \listparindent%
238:        \rightmargin \leftmargin%
239:        \parsep \parskip}\item[]}{\endlist}%
240:
```

```
241:  \newenvironment{example}
242:    {\vspace{1.5\parskip}\begin{small}}
243:    {\end{small}\vspace{1.5\parskip}}
244:
245:  %%------------------------------------------------------------
246:  %% The writing...
247:  %%------------------------------------------------------------
248:  \begin{document}
249:  \begin{titlepage}
250:  \renewcommand*{\thepage}{Title}
251:  \thispagestyle{empty}
252:  \null\vfil\vspace{35pt}
253:  \begin{center}
254:  {\LARGE\bfseries \fomsdt: Paket zur Erstellung\par%
255:                   wissenschaftlicher Arbeiten\par}
256:  \vspace{20pt}
257:  {\Large Axel Kesseler\\*%
258:          \texttt{\small<axel.kesseler@gmx.net>}\par}
259:  \vspace{20pt}
260:  \today\par
261:  \null\vfill
262:  Dieses Dokument wurde aus der Datei \file{\FileName}\\
263:  in der Version~\FileVersion\ vom~\FileDate\ erzeugt.
264:  \end{center}
265:  \null\vfill
266:  \begin{summary}
267:  Das Paket \fomsdt\ ist das Ergebnis einer Diplomarbeit an
268:  der FOM -- Fachhochschule für Oekonomie \& Management in
269:  Essen und soll den Studierenden das Verfassen ihrer
270:  wissenschaftlichen Arbeiten unter Verwendung des Textsatzsystems
271:  \LaTeX\ erleichtern.
272:
273:  \fomsdt\ basiert auf den an der FOM im Fachbereich
274:  Wirtschaftsinformatik anerkannten Regeln zu Erstellung
275:  wissentschaftlicher Arbeiten.
276:  \end{summary}
277:  \end{titlepage}
278:
279:  \tableofcontents
280:
281:  \section{Distribution}
282:
283:  Zum Lieferumfang des Pakets\index{Paket>Umfang} \fomsdt\
284:  gehören die Dateien\index{Paket>Dateien} \file{\fomsdtins},
285:  \file{\fomsdtdtx} und \file{\readmetxt}. Diese Installationsdateien
286:  beinhalten neben den eigentlichen Paketdateien, bestehend aus
287:  eine Klassendatei und mehreren Stildateien, auch die zum Paket
288:  gehörende Anleitung.
289:
290:  \DescFile{\fomsdtins} Die Datei \file{\fomsdtins} beinhaltet
291:  das zur Extraktion des Pakets \fomsdt\ notwendige \LaTeX"~Skript,
```

```
292:  dessen Hauptaufgabe darin besteht, die Aufteilung des Inhalts
293:  der Datei \file{\fomsdtdtx} vorzunehmen.
294:
295:  \DescFile{\fomsdtdtx} Der vollständige Paketcode ist in der
296:  Datei \file{\fomsdtdtx} enthalten. Ebenfalls in dieser Datei
297:  enthalten ist der Code zur Erstellung die hier vorliegende
298:  Paketdokumentation.
299:
300:  \DescFile{\readmetxt} Die Datei \file{\readmetxt} enthält neben
301:  allgemeinen Informationen auch ein Kurzanleitung zur Erstellung
302:  der hier vorliegenden Dokumentation.
303:
304:  Die hieran anschließenden Kapitel gehen sowohl auf die Erstellung
305:  der Dokumentation als auch auf die Extraktion und die Installation
306:  des Pakets \fomsdt\ im \LaTeX"~System ein.
307:
308:  \section{Dokumentation}\label{DocCreation}
309:
310:  Zur Erstellung der Dokumentation wird zunächst die Datei
311:  \file{\FileName} einmal kompiliert. Traten hierbei keine Fehler
312:  auf, wird die Datei \file{\FileName} ein zweites mal kompiliert.
313:  Dies ist notwendig, da erst jetzt alle Indexeinträge auf die
314:  "‘richtigen"’ Seiten verweisen.
315:
316:  Daran anschließend wird der Index generiert. Traten auch hierbei
317:  keine Fehler auf, ist die Datei \file{\FileName} noch ein drittes
318:  und abschließendes mal zu kompilieren.
319:
320:  Der gesamte Prozess zur Erstellung der Paketdokumentation
321:  \index{Paket>Dokumentation} wird durch die Eingabe verschiedener
322:  \LaTeX"~Kommandos\index{Paket>Dokumentation>Kommandos} am Prompt
323:  einer Shell beziehungsweise Konsole durchgeführt. Die nachfolgende
324:  Sequenz zeigt die einzelnen Befehle, mit deren Hilfe die
325:  Dokumentation generiert wird.
326:
327:  \begin{verbatim}
328:   $ pdflatex fomsdt.dtx
329:   $ pdflatex fomsdt.dtx
330:   $ makeindex -s gind.ist -o fomsdt.ind fomsdt.idx
331:   $ pdflatex fomsdt.dtx
332:  \end{verbatim}
333:
334:  Sollten nach dem dritten Kompiliervorgang eventuell noch Fehler
335:  oder Warnungen auftreten, dann empfiehlt es sich, das Kommando
336:  \CommandIndex{pdflatex}{\small| fomsdt.dtx|} erneut anzuwenden,
337:  um die Dokumentation ein viertes mal zu kompilieren.
338:
339:  Nach Beendigung des Kompiliervorganges sollte im aktuellen
340:  Verzeichnis nun auch die Datei \FileIndex{\fomsdtpdf}, neben
341:  weiteren Ausgabedateien, vorhanden sein.
342:
```

```
343:  \section{Installation}
344:
345:  Die Installation des Pakets \fomsdt\ wird in mehreren
346:  Einzelschritten durchgeführt. Dabei muss an einigen Stellen
347:  auf die Besonderheiten der unterschiedlichen Betriebssysteme
348:  Rücksicht genommen werden. Entsprechende Hinweise erfolgen an
349:  den jeweiligen Stellen.
350:
351:  \subsection{Verzeichnisstruktur}\label{DirCreation}
352:
353:  Der erste Schritt der Installation des Pakets \fomsdt\ ist
354:  die Erzeugung der notwendigen Verzeichnisstruktur. Hierfür
355:  werden innerhalb der \LaTeX"~Umgebung insgesamt zwei
356:  Verzeichnisse\index{Installation>Verzeichnisses}, jedes mit
357:  dem Namen \file{fomsdt}, angelegt. Dabei ist zu beachten,
358:  dass diese Verzeichnisse als Unterverzeichnisse der Verzeichnisse
359:  \file{\$TEXROOT/tex/latex} und \file{\$TEXROOT/makeindex}
360:  erstellt werden müssen!
361:
362:  Der Bezeichner \file{\$TEXROOT} dient an dieser Stelle nur
363:  als Platzhalter, da das entsprechende \LaTeX"~Basisverzeichnis
364:  stark vom eingesetzten Betriebssystem abhängt.
365:
366:  \DescOnly{Windows} Unter Windows wird zumeist MiK\TeX\ als
367:  \LaTeX"~Umgebung eingesetzt. Daher ist für \file{\$TEXROOT}
368:  das Verzeichnis der installierten MiK\TeX"~Distribution zu
369:  verwenden\index{Paket>Verzeichnisse>Windows}. Dies könnte zum
370:  Beispiel \file{C:\bs{}Programme\bs{}MiKTeX\bs{}} sein. Somit
371:  ergeben sich für dieses Beispiel und das Betriebssystem Windows
372:  die beiden Verzeichnisse
373:  \file{C:\bs{}Programme\bs{}MiKTeX\bs{}tex\bs{}latex\bs{}fomsdt}
374:  und \file{C:\bs{}Programme\bs{}MiKTeX\bs{}makeindex\bs{}fomsdt}.
375:
376:  \DescOnly{Linux} Unter Linux wird die \LaTeX"~Distribution
377:  zumeist im Verzeichnis \file{/usr/share/texmf/}
378:  installiert\index{Paket>Verzeichnisse>Linux}, sodass als
379:  \file{\$TEXROOT} genau dieser Pfad als Basis zu verwenden ist.
380:  Somit ergeben sich für dieses Beispiel und das Betriebssystem
381:  Linux die beiden vollständigen Pfade
382:  \file{/usr/share/texmf/tex/latex/fomsdt} und
383:  \file{/usr/share/texmf/makeindex/fomsdt}.
384:
385:  \subsection{Kopieren und Entpacken}
386:
387:  Nachdem zuvor die Zielverzeichnisse, wie in
388:  Abschnitt~\ref{DirCreation} beschrieben, erstellt wurden,
389:  werden die zum Paket gehörenden Dateien \FileIndex{\fomsdtins},
390:  \FileIndex{\fomsdtdtx} und \FileIndex{\readmetxt} von ihrer
391:  Quelle, beispielsweise das Verzeichnis, in welchem die Dateien
392:  nach ihrem Download gespeichert wurden, in das Verzeichnis
393:  \file{\$TEXROOT/tex/latex/fomsdt}
```

```
394:  kopiert\index{Paket>Dateien>kopieren}.
395:
396:  Die unter Kapitel~\ref{DocCreation} generierte Datei
397:  \FileIndex{\fomsdtpdf} wird ebenfalls in dieses Verzeichnis
398:  kopiert. Dies weicht zwar von der \LaTeX"~Empfehlung für den
399:  Dokumentenspeicherort ab, dient aber dem Zusammenhalt des
400:  gesamten Pakets.
401:
402:  Jetzt wird im Verzeichnis \file{\$TEXROOT/tex/latex/fomsdt}
403:  eine Shell beziehungsweise Konsole geöffnet und darin der
404:  Befehl zum Entpacken\index{Paket>Dateien>entpacken} des
405:  Pakets wie folgt eingegeben und ausgeführt:
406:
407:  \begin{example}
408:  | $ |\CommandIndex{pdflatex}| fomsdt.ins|
409:  \end{example}
410:
411:  Konnten bis hierhin alle Schritte fehlerfrei ausgeführt werden,
412:  dann sollten im Verzeichnis\index{Dateien>Zielverzeichnis}
413:  \file{\$TEXROOT/tex/latex/fomsdt} mindestens die hier gezeigten
414:  Dateien vorhanden sein.
415:
416:  \begin{center}
417:  \begin{tabular}{ccccc}
418:  \file{\fomsdtdtx} && \file{\fomsdtcls} && \file{\fomfodsty} \\
419:  \file{\fomsdtins} && \file{\fomdsssty} && \file{\fomiscsty} \\
420:  \file{\readmetxt} && \file{\fomdessty} && \file{\fomidxist} \\
421:  \file{\fomsdtpdf} && \file{\fomparsty} &&                   \\
422:  \end{tabular}
423:  \end{center}
424:
425:  Als nächstes ist noch die Datei \file{\fomidxist} in das
426:  Verzeichnis \file{\$TEXROOT/makeindex/fomsdt/} zu
427:  verschieben\index{Paket>Dateien>verschieben}!
428:
429:  \subsection{Aktualisierung der Umgebung}
430:
431:  Als dritter und letzter Schritt der Installation des Pakets
432:  \fomsdt\ ist nun noch die
433:  Aktualisierung\index{Paket>Aktualisierung>Umgebung}
434:  der \LaTeX"~Umgebung durchzuführen. Hierfür wird am Prompt
435:  der noch geöffneten Eingabeaufforderung das
436:  Aktualisierungskommando\index{Paket>Aktualisierung>Kommando}
437:  wie folgt eingegeben:
438:
439:  \begin{example}
440:  | $ |\CommandIndex{texhash}
441:  \end{example}
442:
443:  Speziell für die Benutzer einer MiK\TeX"~Distribution gilt:
444:  Sollte das Kommando \CommandIndex{texhash} \emph{nicht} zur
```

```
445:  gewünschten Aktualisierung des
446:  \LaTeX"~Systems\index{Paket>Aktualisierung} führen, dann sollte
447:  das Aktualisierungskommando der MiK\TeX"~Distribution verwendet
448:  werden. In diesem Fall ist am Prompt stattdessen das Kommando
449:  \CommandIndex{initexmf}{\small| -u|} einzugeben.
450:
451:  Mit der fehlerfreien Registrierung ist nun auch die Installation
452:  des Pakets erfolgreich beendet. Das Paket \fomsdt\ kann von
453:  nun an zusammen mit \LaTeX\ verwendet werden.
454:
455:  \subsection{Kurzanleitung}
456:
457:  Für die Eiligen, die Kurzfassung\index{Installation>Kurzfassung}
458:  der Anleitung zur Installation des Pakets \fomsdt\ im
459:  \LaTeX"~System.
460:
461:  \begin{itemize}
462:  \item Jeweils ein Unterverzeichnis mit dem Namen \file{fomsdt}
463:  \begin{itemize}
464:    \item im Verzeichnis \file{\$TEXROOT/tex/latex} und
465:    \item im Verzeichnis \file{\$TEXROOT/makeindex}
466:  \end{itemize}
467:  erstellen.
468:  \item Die Dateien \file{\fomsdtins}, \file{\fomsdtdtx} und
469:    \file{\readmetxt} in das Verzeichnis
470:    \file{\$TEXROOT/tex/latex/fomsdt} kopieren.
471:  \item Die Paketdokumentation, also die Datei \file{\fomsdtpdf},
472:    ebenfalls in das Verzeichnis \file{\$TEXROOT/tex/latex/fomsdt}
473:    kopieren! Oder, falls noch nicht vorhanden, in diesem
474:    Verzeichnis erst erstellen.
475:  \item Im Verzeichnis \file{\$TEXROOT/tex/latex/fomsdt} eine Shell
476:    öffnen und dort den Befehl
477:    \CommandIndex{pdflatex}~\file{\fomsdtins} eingeben und ausführen.
478:  \item Die nun vorhandene Datei \file{\fomidxist} in das Verzeichnis
479:    \file{\$TEXROOT/makeindex/fomsdt} verschieben!
480:  \item Zur Aktualisierung des \LaTeX"~Systems am Kommandoprompt
481:    noch den Befehl \CommandIndex{texhash} eingeben.
482:  \end{itemize}
483:
484:  Happy \TeX{}ing!
485:
486:  \section{Paketübersicht}
487:
488:  Dieser Abschnitt verschafft einen Überblick über die einzelnen
489:  Komponenten\index{Paket>Übersicht} der \emph{\LaTeX"~Nomenklatur
490:  \fomsdt\ zur Erstellung wissenschaftlicher Arbeiten} und geht
491:  kurz auf die Aufgaben sowie die Bedeutung der einzelnen
492:  Komponenten ein.
493:
494:  \DescFile{\fomsdtcls} Die Datei \file{\fomsdtcls} meint \FOMSDT\
495:  und enthält den Code der Dokumentenklasse des Pakets. Weiterhin
```

```
496:  werden von dieser Paketklassse verschiedene Optionen zur Änderung
497:  des Dokumentenlayouts angeboten. Zusätzlich verfügt diese
498:  Dokumentenklasse auch über Alternativen zur Deckblattgestaltung.
499:
500:  \DescFile{\fomdsssty} Die Datei \file{\fomdsssty} meint \FOMDSS\
501:  und enthält Makros, die es dem Benutzer erlauben, den einleitenden
502:  Teil einer wissenschaftlichen Arbeit mit leichter Hand zu
503:  erstellen. Beispielsweise beinhaltet der Einleitungsteil das
504:  Inhaltsverzeichnis und ein Abbildungsverzeichnis. Weitere
505:  Verzeichnisse werden ebenfalls unterstützt.
506:
507:  \DescFile{\fomdessty} Die Datei \file{\fomdessty} meint \FOMDES\
508:  und enthält Makros für den abschließenden Teil der
509:  wissenschaftlichen Arbeit. Beispielsweise beinhaltet der
510:  Dokumentenabschluss den Index, das Glossar und ähnliches.
511:
512:  \DescFile{\fomparsty} Die Datei \file{\fomparsty} meint \FOMPAR\
513:  und enthält spezielle aber auch nützliche Makros zur Formatierung
514:  von Absätzen. Beispielsweise können hierdurch beidseitig
515:  eingezogene Absätze oder auch einseitig eingezogene Absätze
516:  mit einem Betreffbezeichner erzeugt werden.
517:
518:  \DescFile{\fomfodsty} Die Datei \file{\fomfodsty} meint \FOMFOD\
519:  und enthält Makros zur Einbindung von Abbildungen, zur Gestalltung
520:  von Tabellen und zur Erzeugung von Codelistings. Der Vorteil
521:  dieser Definitionsdatei besteht in der Einfachheit der Anwendung
522:  der bereitgestellten Makros.
523:
524:  \DescFile{\fomiscsty} Die Datei \file{\fomiscsty} meint \FOMISC\
525:  und dient als Sammelbecken für weitere nützliche Makros, die
526:  vom Sinnzusammenhang in keiner anderen Definitionsdatei
527:  untergebracht werden können.
528:
529:  \DescFile{\fomidxist} Die Datei \file{\fomidxist} meint \FOMIDX\
530:  und enthält verschiedene Konfigurationen für eine ansprechendere
531:  Gestaltung des Indexes. Sie wird durch das Programm
532:  \CommandIndex{make\-in\-dex} verwendet und ist daher nur indirekt
533:  verwendbar. Zudem enthält diese Datei keine Makros im klassischen
534:  Sinne. Aus diesem Grund wird innerhalb dieser Dokumentation
535:  auf die weiterführende Beschreibung der Datei \file{\fomidxist}
536:  verzichtet.
537:
538:  \DescOnly{Mehrsprachigkeit} Das Paket \fomsdt\ unterstützt
539:  derzeit noch nicht die Verwendung unterschiedlicher Sprachen.
540:  Somit ist der Benutzer aufgefordert, bei einer von Deutsch
541:  verschiedenen Sprache, die Standardwerte der jeweiligen
542:  Makros zu überschreiben.
543:
544:  \section{Dokumentenklasse}
545:
546:  Die Dokumentenklasse \fomsdt, implemetiert in der Datei
```

```
547:  \FileIndex{\fomsdtcls}, basiert auf der \LaTeX"~eigenen Klasse
548:  \DependsIndex{article}. Sie verfügt somit über alle Optionen
549:  ihrer Basisklasse. Weiterhin verfügt \fomsdt\ über eigene
550:  Optionen, die in diesem Abschnitt einzeln beschrieben werden.
551:
552:  \DescMacro{\documentclass} Verwendet wird die Klasse \fomsdt\
553:  analog zur Verwendung aller anderen \LaTeX"~Dokumentenklassen.
554:  Daher ist die erste Zeile der Hauptdatei eines neuen Dokuments
555:  basierend auf der Klasse \fomsdt\ die Zeile
556:  {\small|\documentclass|\oarg{Optionen}|{fomsdt}|}.
557:
558:  Die Angabe mehrerer Parameter im Optionsteil des Kommandos
559:  {\small|\documentclass|} ist möglich, wobei einzelne Optionen
560:  durch ein Komma voneinander getrennt werden.
561:
562:  An dieser Stelle noch ein Kurzbeispiel\index{Beispiel>Hallo Welt}
563:  zu Illustration der zuvor beschriebenen Details zur Verwendung
564:  der Klasse \fomsdt.
565:
566:  \begin{verbatim}
567:   \documentclass[nochapter,noheadrule]{fomsdt}
568:   \begin{document}
569:    Hallo Welt!
570:   \end{document}
571:  \end{verbatim}
572:
573:  \subsection{Optionen}
574:
575:  Die Klasse \fomsdt\ verfügt über Optionen, die vornehmlich
576:  zur Steuerung des Layouts eines Dokuments dienlich sind.
577:  Diese Optionen werden im Folgenden näher beschrieben.
578:
579:  \DescOption{moderntitle} Die wissenschaftliche Arbeit soll
580:  über ein Titelblatt\index{Titelblatt} mit "‘modernem"’ Aussehen
581:  verfügen. Als Standard für den Stil des Titels wird
582:  \defval{kein Stil} verwendet.
583:
584:  \DescOption{classictitle} Die wissenschaftliche Arbeit soll
585:  über ein Titelblatt\index{Titelblatt} mit "‘klassischem"’ Aussehen
586:  verfügen. Als Standard für den Stil des Titels wird
587:  \defval{kein Stil} verwendet.
588:
589:  Wird als Stil des Titelblatts weder die Option \option{moderntitle}
590:  noch die Option \option{classictitle} angegeben, dann können
591:  zur Erzeugung des Titels weiterhin die \LaTeX"~eigenen Makros
592:  verwendet werden.
593:
594:  Werden hingegen beide Stiloptionen des Titelblatts gemeinsam
595:  verwendet, die Reihenfolge ist dabei unerheblich, dann wird
596:  als Standard der klassische Titelstil verwendet.
597:
```

```
598:  \DescOption{lockflag} Diese Option schaltet den
599:  Sperrvermerks\index{Sperrvermerk} auf der Seite direkt hinter
600:  der Titelseite ein. Der Standardwert für die Verwendung des
601:  Sperrvermerks ist \defval{aus}.
602:
603:  Die Verwendung des Sperrvermerks ist jedoch nur bei den
604:  wissenschaftlichen Arbeiten notwendig, die der Öffentlichkeit
605:  unzugänglich sein sollen. Dies ist beispielsweise häufig bei
606:  Diplomarbeiten der Fall, da sie eventuell firmeninterne Daten
607:  enthalten.
608:
609:  \DescOption{affidavit} Mit Hilfe diese Option wird die
610:  eidesstattliche Versicherung\index{Versicherung (eidesstattlich)}
611:  des Autos auf der letzten Seite der wissenschaftlichen Arbeit
612:  aktiviert. Der Standard für die Verwendung der eidesstattlichen
613:  Versicherung ist \defval{aus}.
614:
615:  Verwendet wird die eidesstattlichen Versicherung in der Regel
616:  nur bei Diplomarbeiten. Daher kann bei Studienarbeiten,
617:  Hausarbeiten oder Fallstudien grundsätzlich auf die eidesstattliche
618:  Versicherung verzichtet werden.
619:
620:  \DescOption{noheadrule} Diese Option schaltet die Linie zwischen
621:  der Kopfzeile und dem restlichen Text aus. Der Standardwert
622:  für die Verwendung dieser Trennlinie ist \defval{an}.
623:
624:  \DescOption{nochapter} Diese Option schaltet die Anzeige des
625:  Kapitels in der Kopfzeile aus. Der Standardwert für die Verwendung
626:  des Kapitels in der Kopfzeile ist \defval{an}.
627:
628:  \DescOption{coloredlinks} Die wissenschaftliche Arbeit zeigt
629:  Hyperlinks in PDF"~Aus\-gabedateien in \emph{blau}. Der
630:  Standardwert für die Hyperlinkfarbe ist \defval{schwarz}.
631:
632:  \subsection{Makros}
633:
634:  Werden in den Klassenoptionen entweder \option{moderntitle} oder
635:  \option{classictitle} als Stil für die Titelseite angegeben, dann
636:  ist die Angabe weiterer Informationen unumgänglich. Hierfür werden
637:  von \fomsdt\ entsprechende Makros bereitgestellt.
638:
639:  \DescMacro{\General} Dieser Wert legt die Art der wissenschaftlichen
640:  Ausarbeitung fest. Es handelt sich hierbei um einen optionalen
641:  Parameter. Der Standardwert ist \defval{Hausarbeit}.
642:
643:  \DescMacro{\Date} Dieser Wert gibt das Datum der wissenschaftlichen
644:  Ausarbeitung an. Es handelt sich hierbei um einen optionalen
645:  Parameter. Der Standardwert ist \defval{Essen, den \bs{}today}.
646:
647:  \DescMacro{\Institute} Dieser Wert legt fest, an welcher Hochschule
648:  die wissenschaftliche Ausarbeitung geschrieben wurde. Es handelt
```

```
649:    sich hierbei um einen optionalen Parameter. Der Standardwert ist
650:    \defval{FOM -- Fachhochschule für Oekonomie \& Management Essen}.
651:
652:    \DescMacro{\Course} Dieser Wert definiert den Studiengang, für
653:    den die wissenschaftliche Ausarbeitung geschrieben wurde. Es
654:    handelt sich hierbei um einen optionalen Parameter. Der
655:    Standardwert ist \defval{Berufsbegleitender Studiengang}.
656:
657:    \DescMacro{\Semester} Dieser Wert gibt das Semester an, in dem
658:    die wissenschaftliche Ausarbeitung erstellt wurde. Es handelt
659:    sich hierbei um einen optionalen Parameter. Der Standardwert
660:    ist \defval{\holder{nichts}}.
661:
662:    \DescMacro{\Subject} Dieser Pflichtwert gibt entweder das Fach
663:    oder das Thema der wissenschaftlichen Ausarbeitung an.
664:
665:    \DescMacro{\Title} Dieser Pflichtwert definiert den Titel der
666:    wissenschaftlichen Ausarbeitung.
667:
668:    \DescMacro{\Professor} Dieser Pflichtwert legt den Betreuer des
669:    Studierenden fest. Das Makro verfügt sowohl über einen optionalen
670:    Parameter als auch über einen Pflichtparameter. Die Verwendung
671:    erfolgt als {\small|\Professor|\oarg{Label}\marg{Name}}. Der
672:    Standardwert für den optionalen Parameter \holder{Label} ist
673:    \defval{Betreuer:}.
674:
675:    \DescMacro{\Student} Dieser Pflichtwert legt den Autor der
676:    wissenschaftlichen Ausarbeitung fest. Das Makro verfügt sowohl über
677:    einen optionalen Parameter als auch über einen Pflichtparameter. Die
678:    Verwendung erfolgt als {\small|\Student|\oarg{Label}\marg{Name}}. Der
679:    Standardwert für den optionalen Parameter \holder{Label} ist
680:    \defval{Autor:}.
681:
682:    An dieser Stelle zur Illustration ein Beispiel zur Verwendung der
683:    Parameter für die Titelseite\index{Beispiel>Titelblatt}.
684:
685:    \begin{verbatim}
686:     \General{Hausarbeit}
687:     \Date{Stadt, den \today}
688:     \Institute{FOM}
689:     \Course{Berufsbegleitendes Studium}
690:     \Semester{6. Semester}
691:     \Subject{Hausarbeit}
692:     \Title{Betriebssysteme}
693:     \Professor[Betreuer:]{Prof. Karl Muster}
694:     \Student[Autor:]{Karin Muster (291067)}
695:     \maketitle
696:    \end{verbatim}
697:
698:    Wird zur Erstellung der Arbeit das Programm \CommandIndex{pdflatex}
699:    verwendet, dann besteht die Möglichkeit, zusätzliche Angaben über
```

```
700:  das erstellte Dokument in den PDF"~Eigenschaften zu hinterlegen.
701:
702:  Wird hingegen das Programm \CommandIndex{latex} anstelle von
703:  \CommandIndex{pdflatex} zur Erstellung der Arbeit verwendet, dann
704:  werden eventuell gesetzte PDF"~Eigenschaften ignoriert.
705:
706:  In jedem Fall ist bei der Verwendung der im Folgenden beschriebenen
707:  Makros zu beachten, dass sie in der Präambel, also noch vor der Zeile
708:  {\small|\begin{document}|}, zu verwenden sind!
709:
710:  \DescMacro{\PDFTitle} Dieser Wert setzt die Eigenschaft \emph{Titel}
711:  der PDF"~Aus\-ga\-be\-da\-tei. Der Standardwert ist
712:  \defval{\holder{leer}}.
713:
714:  \DescMacro{\PDFAuthor} Dieser Wert setzt die Eigenschaft \emph{Autor}
715:  der PDF"~Aus\-ga\-be\-da\-tei. Der Standardwert ist
716:  \defval{\holder{leer}}.
717:
718:  \DescMacro{\PDFSubject} Dieser Wert  setzt die Eigenschaft
719:  \emph{Thema} der PDF"~Aus\-ga\-be\-da\-tei. Der Standardwert
720:  ist \defval{\holder{leer}}.
721:
722:  \DescMacro{\PDFProducer} Dieser Wert setzt die Eigenschaft
723:  \emph{Erzeuger} der PDF"~Aus\-ga\-be\-da\-tei. Der Standardwert
724:  ist \defval{PDFLaTeX}.
725:
726:  Zur Illustration hier ein Beispiel\index{Beispiel>PDF"~Eigenschaften}
727:  zur Verwendung der zuvor genannten Makros.
728:
729:  \begin{verbatim}
730:   \PDFTitle{Betriebssysteme}
731:   \PDFAuthor{Karin Muster}
732:   \PDFSubject{Hausarbeit}
733:   \PDFProducer{MiKTeX}
734:  \end{verbatim}
735:
736:  Soll in einer wissenschaftlichen Ausarbeitung ein Sperrvermerk
737:  verwendet werden, dann ist die Angabe weiterer Informationen
738:  notwendig. Hierfür werden von \fomsdt\ die im Folgenden beschriebenen
739:  Makros zur Verfügung gestellt.
740:
741:  \DescMacro{\LockflagContent} Mit Hilfe des Makros
742:  {\small|\LockflagContent|\oarg{Signatur}\marg{Inhalt}} ist es möglich,
743:  den Inhalt des Sperrvermerks zu beeinflussen. Hierbei bestimmt der
744:  Pflichtparameter \marg{Inhalt} den Text\index{Sperrvermerk>Text}
745:  Sperrvermerks wie folgt:
746:
747:  \OptExplain{\option{std}}{Wird {\small|\bs{}LockflagContent\{std\}|}
748:  verwendet, dann enthält der Sperrvermerk einen Standardtext, der
749:  bereits zum Zeitpunkt des Ladens der Dokumentenklasse \fomsdt\
750:  voreingestellt wird.}
```

```
751:
752:  \OptExplain{\option{fom}}{Wird {\small|\bs{}LockflagContent\{fom\}|}
753:  verwendet, dann enthält der Sperrvermerk einen FOM"~spezifischen
754:  Text.}
755:
756:  \OptExplain{\option{\dots}}{Wird weder \option{std} noch \option{fom}
757:  als Parameter verwendet, dann enthält der Sperrvermerk den in
758:  {\small|\{\dots\}|} angegebenen und somit frei wählbaren Text.}
759:
760:  Der optionale Parameter \oarg{Signatur} erlaubt das explizite
761:  Einschalten der Signatur\index{Sperrvermerk>Signatur} des
762:  Sperrvermerks wie folgt:
763:
764:  \OptExplain{\option{sig}}{Wird als optionaler Parameter \option{sig}
765:  verwendet, dann enthält der Sperrvermerk eine aus den Feldern
766:  {\small|(Ort, Datum)|} und {\small|(Eigenhändige Unterschrift)|}
767:  bestehende Signatur.}
768:
769:  \DescMacro{\LockflagDate} Dieses Makro wird verwendet, um Ort und
770:  Datum der Signatur des Sperrvermerks festzulegen. Hierfür ist jedoch
771:  die Angabe des optionalen Parameters \oarg{Signatur} des Markos
772:  {\small|\LockflagContent|} notwendig. Anderenfalls hat die Verwendung
773:  von {\small|\LockflagDate|} keine Auswirkung. Der Standardwert für
774:  {\small|\LockflagDate|} ist \defval{\holder{leer}}.
775:
776:  \DescMacro{\LockflagCompany} Dieses Makro wird verwendet, um den Namen
777:  der Firma festzulegen, der innerhalb des Sperrvermerks zu verwenden
778:  ist.  Der Standardwert ist \defval{???}.
779:
780:  \DescOnlyNoIdx{Achtung!} Die Verwendung des Sperrvermerks verursacht
781:  Abhängigkeiten\index{Sperrvermerk>Abhängigkeit} zu anderen
782:  \fomsdt"~Makros, die an dieser Stelle kurz zusammengefasst werden:
783:
784:  \begin{itemize}
785:  \item Wird {\small|\LockflagContent{fom}|} verdendet, dann werden
786:  durch den FOM"~spezifischen Sperrvermerk die Inhalte der Makros
787:  \MacroIndex{\General}, \MacroIndex{\Title} und
788:  \MacroIndex{\LockflagCompany} verwendet!
789:
790:  \item Wird {\small|\LockflagContent{std}|} verdendet, dann werden
791:  durch den Standardsperrvermerk die Inhalte der Makros
792:  \MacroIndex{\General} und \MacroIndex{\LockflagCompany} verwendet!
793:
794:  \item Wird {\small|\LockflagContent|\marg{freier Text}} verdendet,
795:  dann besteht keine Abhängigkeit zu anderen Makros!
796:
797:  \item Wird {\small|\LockflagContent[sig]|\marg{Inhalt}} verdendet,
798:  dann wird durch die Signatur des Sperrvermerks der Inhalt des Makros
799:  \MacroIndex{\LockflagDate} verwendet!
800:  \end{itemize}
801:
```

```
802: \DescMacro{\LockflagHeader} Dieses Makro dient der Änderung
803: der Überschrift des Sperrvermerks. Hierfür wird im Parameter
804: \marg{Überschrift} der geänderte Text der Überschrift angegeben.
805: Der Standardwert für {\small|\LockflagHeader|} ist
806: \defval{\holder{Sperrvermerk}}.
807:
808: \DescMacro{\LockflagSigLeft} Dieses Makro wird verwendet, um den
809: linken Teil der Signatur des Sperrvermerks zu ändern. Hierfür wird
810: im Parameter \marg{lin\-ker Sig\-na\-tur\-teil} der geänderte Text
811: des linken Signatur\-teils angegeben. Der Standardwert für
812: {\small|\LockflagSigLeft|} ist \defval{\holder{(Ort, Datum)}}.
813:
814: \DescMacro{\LockflagSigRight} Dieses Makro wird verwendet, um den
815: rechten Teil der Sig\-na\-tur des Sperrvermerks zu ändern. Hierfür
816: wird im Parameter \marg{rech\-ter Sig\-na\-tur\-teil} der geänderte
817: Text des rechten Sig\-na\-tur\-teils angegeben. Der Standardwert
818: für {\small|\LockflagSigRight|} ist \defval{\holder{(Eigenhändige
819: Unterschrift)}}.
820:
821: \DescMacro{\AffidavitContent}
822: Das Makro {\small|\AffidavitContent|\oarg{Signatur}\marg{Inhalt}}
823: ermöglicht es, den Inhalt der eidesstattlichen Versicherung zu
824: beeinflussen. Hierbei bestimmt der Pflichtparameter \marg{Inhalt} den
825: Text\index{Versicherung (eidesstattlich)>Text} der eidesstattlichen
826: Versicherung wie folgt:
827:
828: \OptExplain{\option{std}}{Wird {\small|\bs{}AffidavitContent\{std\}|}
829: verwendet, dann enthält die eidesstattliche Versicherung einen
830: Standardtext, der bereits zum Zeitpunkt des Ladens der
831: Dokumentenklasse \fomsdt\ voreingestellt wird.}
832:
833: \OptExplain{\option{\dots}}{Wird der Parameter \option{std} nicht
834: verwendet, dann enthält die eidesstattliche Versicherung den in
835: {\small|\{\dots\}|} angegebenen und somit frei wählbaren Text.}
836:
837: Der optionale Parameter \oarg{Signatur} erlaubt das explizite
838: Einschalten der Signatur\index{Versicherung (eidesstattlich)>Signatur}
839: der eidesstattlichen Versicherung wie folgt:
840:
841: \OptExplain{\option{sig}}{Wird als optionaler Parameter \option{sig}
842: verwendet, dann enthält die eidesstattliche Versicherung eine aus den
843: Feldern {\small|(Ort, Datum)|} und {\small|(Eigenhändige Unterschrift)|}
844: bestehende Signatur.}
845:
846: \DescMacro{\AffidavitDate} Dieses Makro wird verwendet, um Ort und
847: Datum der Signatur der eidesstattlichen Versicherung festzulegen.
848: Hierfür ist jedoch die Angabe des optionalen Parameters
849: \oarg{Signatur} des Markos {\small|\AffidavitContent|} notwendig.
850: Anderenfalls hat die Verwendung von {\small|\AffidavitDate|} keine
851: Auswirkung. Der Standardwert ist \defval{\holder{leer}}.
852:
```

```
853:  \DescOnlyNoIdx{Achtung!} Die Verwendung der eidesstattlichen
854:  Versicherung verursacht
855:  Abhängigkeiten\index{Versicherung (eidesstattlich)>Abhängigkeit}
856:  zu anderen \fomsdt"~Makros, die an dieser Stelle kurz zusammengefasst
857:  werden:
858:
859:  \begin{itemize}
860:  \item Wird {\small|\AffidavitContent{std}|} benutzt, dann wird von
861:  der eidesstattlichen Versicherung der Inhalt des Makros
862:  \MacroIndex{\General} verwendet!
863:
864:  \item Wird {\small|\AffidavitContent|\marg{freier Text}} benutzt,
865:  dann besteht keine Abhängigkeit zu anderen Makros!
866:
867:  \item Wird {\small|\AffidavitContent[sig]|\marg{Inhalt}} benutzt,
868:  dann wird durch die Signatur der eidesstattlichen Versicherung der
869:  Inhalt des Makros \MacroIndex{\AffidavitDate} verwendet!
870:  \end{itemize}
871:
872:  \DescMacro{\AffidavitHeader} Dieses Makro dient der Änderung der
873:  Überschrift der eidesstattlichen Versicherung. Hierfür wird im
874:  Parameter \marg{Ü\-ber\-schrift} der geänderte Text der Überschrift
875:  angegeben. Der Standardwert für {\small|\AffidavitHeader|} ist
876:  \defval{\holder{Eh\-ren\-wört\-liche Er\-klä\-rung}}.
877:
878:  \DescMacro{\AffidavitSigLeft} Dieses Makro wird verwendet, um den
879:  linken Teil der Sig\-na\-tur der eidesstattlichen Versicherung zu
880:  ändern. Hierfür wird im Parameter \marg{linker Signatur\-teil} der
881:  geänderte Text des linken Signatur\-teils angegeben. Der Standardwert
882:  für {\small|\AffidavitSigLeft|} ist \defval{\holder{(Ort, Datum)}}.
883:
884:  \DescMacro{\AffidavitSigRight} Dieses Makro wird verwendet, um den
885:  rechten Teil der Sig\-na\-tur der eidesstattlichen Versicherung zu
886:  ändern. Hierfür wird im Parameter \marg{rechter Signatur\-teil} der
887:  geänderte Text des rechten Signatur\-teils angegeben. Der Standardwert
888:  für {\small|\AffidavitSigRight|} ist \defval{\holder{(Eigenhändige
889:  Unterschrift)}}.
890:
891:  Zur Unterstützung der Verständlichkeit der zuvor beschriebenen
892:  Zusammenhänge hier noch einige Beispiele zur Verwendung des
893:  Sperrvermerks und der eidesstattlichen Versicherung.
894:
895:  \index{Beispiel>Sperrvermerk}\begin{verbatim}
896:   % Ein Dokument mit Titelseite und Sperrvermerk für
897:   % wissenschaftliche Arbeiten aber OHNE eidesstattliche
898:   % Versicherung. Der Sperrvermerk 'fom', der Inhalt von
899:   % \General und die Signatur inklusive Signaturdatum
900:   % werden verwendet.
901:   \documentclass[lockflag,classictitle]{fomsdt}
902:   \begin{document}
903:   \General{Fallstudie}
```

```
904:     \Title{Betriebssysteme}
905:     \Professor{Prof. Karl Muster}
906:     \Student{Karin Muster (291067)}
907:     \LockflagDate{Essen, den \today}
908:     \LockflagCompany{Müller GmbH \& Co KG}
909:     \LockflagContent[sig]{fom}
910:     \maketitle% Sperrvermerk wird erzeugt!
911:     Der Text des Dokuments ...
912:     \end{document}
913:    \end{verbatim}
914:
915:    \index{Beispiel>Versicherung (eidesst.)}\vspace{2\parskip}
916:    \begin{minipage}{\textwidth}
917:    \begin{verbatim}
918:     % Ein Dokument mit Titelseite und eidesstattlicher
919:     % Versicherung für wissenschaftliche Arbeiten aber
920:     % OHNE Sperrvermerk. Die eidesstattliche Versicherung
921:     % verwendet einen freien Text ohne Signatur.
922:     \documentclass[affidavit,classictitle]{fomsdt}
923:     \AffidavitContent{Hiermit versichere ich ...}
924:     \begin{document}
925:     \General{Fallstudie}
926:     \Title{Betriebssysteme}
927:     \Professor{Prof. Karl Muster}
928:     \Student{Karin Muster (291067)}
929:     \maketitle
930:     Der Text des Dokuments ...
931:     \end{document}% Eidesst. Versicherung wird erzeugt!
932:    \end{verbatim}
933:    \end{minipage}\vspace{2\parskip}
934:
935:    \section{Dokumentenbeginn}
936:
937:    Jede wissenschaftliche Arbeit beginnt\index{Dokumentenbeginn} mit
938:    einer Reihe von Verzeichnissen, welche die Struktur der Arbeit
939:    widerspiegeln. Die Bereitstellung dieser Verzeichnisse ist Aufgabe
940:    von \fomdss, welche in der Datei \FileIndex{\fomdsssty} implementiert
941:    ist.
942:
943:    \DescMacro{\usepackage}
944:    Die Verwendung des Stils \fomdss\ erfolgt analog der Verwendung
945:    aller anderen \LaTeX"~Stildateien. Somit wird in der Präambel
946:    der Hauptdatei eines neuen Dokuments die Zeile
947:    {\small|\usepackage|\oarg{Optionen}|{fomdss}|} eingetragen.
948:
949:    Die Angabe mehrerer Parameter im Optionsteil des Kommandos
950:    {\small|\usepackage|} ist möglich, wobei einzelne Optionen durch
951:    ein Komma voneinander getrennt werden.
952:
953:    Zur Illustration ein
954:    Beispieldokument\index{Beispiel>Inhaltsverzeichnis}, das ein
```

```
955:  Inhaltsverzeichnis mit einer Haupt"~ und zwei Unterüberschriften
956:  erzeugt. Der Bezeichner des Inhaltsverzeichnisses wird ebenfalls
957:  mit in das generierte Inhaltsverzeichnis eingetragen.
958:
959:  \begin{verbatim}
960:   \documentclass{fomsdt}
961:   \usepackage[toctotoc]{fomdss}
962:   \begin{document}
963:   \PrintContents
964:   \section{Überschrift 1}
965:   \subsection{Überschrift 1.1}
966:   \subsection{Überschrift 1.2}
967:   \end{document}
968:  \end{verbatim}
969:
970:  \subsection{Optionen}
971:
972:  Zur Steuerung des Layouts der verwendeten Verzeichnisse verfügt
973:  \fomdss\ über die im Folgenden gezeigten Optionen. Bei unbekannten
974:  oder nicht unterstützten Optionen wird eine Warnmeldung ausgegeben,
975:  die zur Information des Benutzers dient.
976:
977:  \DescOption{toctotoc} Diese Option fügt die Überschrift des
978:  Inhaltsverzeichnisses mit in das Inhaltsverzeichnis selbst ein. Der
979:  Standardwert ist \defval{nicht einfügen}.
980:
981:  \DescOption{nopagebreak} Diese Option verhindert den erzwungenen
982:  Seitenwechsel nach jeweils dem Abkürzungs"~, dem Abbildungs"~ und
983:  dem Tabellenverzeichnis. Sie kann besonders bei kurzen Verzeichnissen
984:  nützlich sein. Dies gilt jedoch nicht für das Inhaltsverzeichnis!
985:  Der Standardwert ist \defval{Seitenumbruch}.
986:
987:  \DescOption{abbreviation} Diese Option legt fest, ob das
988:  Abkürzungsverzeichnis überhaupt erzeugt werden soll. Dies ist
989:  notwendig, da das Abkürzungsverzeichnis mit Hilfe des Pakets
990:  \DependsIndex{nomencl} erstellt wird, das standardmäßig \emph{nicht}
991:  installiert ist! Bei Verwendung dieser Option wird auf der Ausgabe
992:  eine Meldung angezeigt, wie unter Verwendung des Programms
993:  \CommandIndex{makeindex} das Abkürzungsverzeichnis erstellt werden
994:  muss. Der Standardwert ist \defval{nicht verwenden}.
995:
996:  \DescOption{abbrevrefpage} Diese Option legt fest, ob im
997:  Abkürzungsverzeichnis eine Seitenreferenz für das jeweils erste
998:  Auftreten einer Abkürzung innerhalb des Textes enthalten sein soll.
999:  Sie ist aber nur dann wirksam, wenn \fomdss\ ebenfalls mit der Option
1000: \option{abbreviation} geladen wird. Anderenfalls hat die Option
1001: \option{abbrevrefpage} keine Auswirkungen. Als Standardwert wird
1002: \defval{keine Seitenreferenz} im Abkürzungsverzeichnis verwendet.
1003:
1004: \subsection{Makros}
1005:
```

```
1006:  Ungeachtet der Ladeoptionen werden von \fomdss\ zur Erstellung
1007:  verschiedener Verzeichnisses die hier gezeigten Makros zur Verfügung
1008:  gestellt.
1009:
1010:  \DescMacro{\PrintContents} Dieses Makro dient der Erzeugung des
1011:  Inhaltsverzeichnisses\index{Inhaltsverzeichnis}. Es verfügt über einen
1012:  optionalen Parameter zur Anpassung des Titels dieses Verzeichnisses.
1013:  Als Standardwert wird \defval{Inhalt} verwendet.
1014:
1015:  \DescMacro{\PrintFigures} Dieses Makro dient der Erzeugung des
1016:  Abbildungsverzeichnisses\index{Abbildungsverzeichnis}. Es verfügt
1017:  über einen optionalen Parameter zur Anpassung des Titels dieses
1018:  Verzeichnisses. Als Standardwert wird \defval{Abbildungen} verwendet.
1019:
1020:  \DescMacro{\PrintTables} Dieses Makro dient der Erzeugung des
1021:  Tabellenverzeichnisses\index{Tabellenverzeichnis}. Es verfügt
1022:  über einen optionalen Parameter zur Anpassung des Titels dieses
1023:  Verzeichnisses. Als Standardwert wird \defval{Tabellen} verwendet.
1024:
1025:  Wurde \fomdss\ zuvor mit der Option \option{abbreviation}
1026:  geladen, dann stehen spezielle Makros zur Erzeugung des
1027:  Abkürzungsverzeichnisses bereit.
1028:
1029:  \DescMacro{\PrintAbbrev} Dieses Makro dient der Erzeugung des
1030:  Abkürzungsverzeichnisses\index{Abkürzungsverzeichnis}. Es verfügt
1031:  über einen optionalen Parameter zur Anpassung des Titels dieses
1032:  Verzeichnisses. Als Standardwert wird \defval{Abkürzungen} verwendet.
1033:
1034:  \DescMacro{\Abbrev} Mit Hilfe des Makros
1035:  {\small|\Abbrev|\marg{Abkürzung}\marg{Beschreibung}} werden innerhalb
1036:  des normalen Textes die Passagen markiert, die später in das
1037:  Abkürzungsverzeichnis übernommen werden sollen. Dabei beinhaltet der
1038:  erste Parameter die Abkürzung selbst und der zweite Parameter enthält
1039:  die Bedeutung der im ersten Parameter angegebenen Abkürzung. Siehe
1040:  hierzu nachfolgendes Beispiel.
1041:
1042:  \DescMacro{\Mark} Das Makro {\small|\Mark|\marg{Buchstabe}} dient der
1043:  Hervorhebung der relevanten Buchstaben innerhalb der Beschreibung
1044:  einer Abkürzung. Alle hierdurch eingefassten Buchstaben werden später
1045:  im Abkürzungsverzeichnis unterstrichen dargestellt. Siehe hierzu
1046:  nachfolgendes Beispiel.
1047:
1048:  Zur Illustration und zur Verbesserung des Verständnisses der
1049:  Verwendung der Makros {\small|\Abbrev|} und {\small|\Mark|}
1050:  an dieser Stelle das bereits angekündigte
1051:  Beispiel\index{Beispiel>Abkürzungsverzeichnis}.
1052:
1053:  \begin{verbatim}
1054:   \documentclass{fomsdt}
1055:   \usepackage[abbreviation]{fomdss}
1056:   \begin{document}
```

```
1057:    \PrintAbbrev
1058:    \Abbrev{IC} {\Mark{I}ntegrated \Mark{C}ircuit}
1059:    \Abbrev{PIN}{\Mark{P}ersonal \Mark{I}dentification
1060:                 \Mark{N}umber}
1061:    \end{document}
1062:   \end{verbatim}
1063:
1064:   Der dargestellt Codeschnipsel bewirkt, dass im Abkürzungsverzeichnis
1065:   die beiden Einträge \defval{IC} und \defval{PIN} wie folgt hinzugefügt
1066:   werden:
1067:
1068:   \begin{list}{}{
1069:       \setlength{\leftmargin}{.2\textwidth}
1070:       \setlength{\labelwidth}{\leftmargin}
1071:       \setlength{\labelsep}{0pt}
1072:       \renewcommand{\makelabel}[1]{\textbf{#1}\ \dotfill\space}}
1073:       \item[IC]{\uline{I}ntegrated \uline{C}ircuit}
1074:       \item[PIN]{\uline{P}ersonal \uline{I}dentification \uline{N}umber}
1075:   \end{list}
1076:
1077:   \DescOnlyNoIdx{Hinweis\index{Abkürzungsverzeichnis>Hinweis}} Für die
1078:   Darstellung von Abkürzungen, Punkte inklusive, wird im
1079:   Abkürzungsverzeichnis die fest eingestellte Breite von 20\% der
1080:   aktuellen Textbreite verwendet. In den meisten Fällen ist dies auch
1081:   völlig ausreichend. Sollte die voreingestellte Standardbreite jedoch
1082:   einmal nicht ausreichend sein, kann sie mit Hilfe des Kommandos
1083:
1084:   \begin{example}
1085:   | \setlength{\nomlabelwidth}{|\holder{Breite}|}|
1086:   \end{example}
1087:
1088:   neu eingestellt werden. Diese Anpassung kann sowohl in der Präambel
1089:   als auch direkt vor Aufruf des Makros \MacroIndex{\PrintAbbrev}
1090:   erfolgen.
1091:
1092:   Die Änderung des Layouts der im Abkürzungsverzeichnis eingetragenen
1093:   Abkürzungen ist ebenfalls möglich. Hierfür wird dann das folgende
1094:   Kommando verwendet:
1095:
1096:   \begin{example}
1097:   | \renewcommand{\nomlabel}[1]{|\holder{Formatierung}|}|
1098:   \end{example}
1099:
1100:   Möchte man zum Beispiel die Abkürzungen des Abkürzungsverzeichnisses
1101:   als schräg gestellten Text und ohne Füllpunkte darstellen, dann kann
1102:   dies durch Anwendung des Kommandos
1103:
1104:   \begin{example}
1105:   | \renewcommand{\nomlabel}[1]{\textsl{#1}}|
1106:   \end{example}
1107:
```

```
1108:  erreicht werden. Auch hierbei gilt, dass die Anpassung entweder in der
1109:  Präambel oder direkt vor Aufruf des Makros \MacroIndex{\PrintAbbrev}
1110:  erfolgen sollte.
1111:
1112:  Unter Umständen kann es sinnvoll sein, die im Abkürzungsverzeichnis
1113:  verwendete Standardseitenreferenz zu ändern. Soll nun beispielsweise
1114:  die Seitenreferenz in \defval{, page \holder{Seite}} geändert werden,
1115:  dann kann dies durch die Anwendung des folgenden Kommandos erfolgen:
1116:
1117:  \begin{example}
1118:  | \renewcommand{\pagedeclaration}[1]{, page~#1}|
1119:  \end{example}
1120:
1121:  Wie oben bereits angesprochen, wird zur Erzeugung des
1122:  Abkürzungsverzeichnisses ein spezielles \LaTeX"~Paket verwendet.
1123:  Dieses Paket wird \emph{nur} zusammen mit dem Programm
1124:  \CommandIndex{makeindex}, in Verbindung mit speziellen
1125:  Aufrufparametern, verwendet. Daher an dieser Stelle die Syntax zur
1126:  Verwendung von \file{makeindex} zur Erstellung des
1127:  Abkürzungsverzeichnisses.
1128:
1129:  \begin{example}
1130:  | $ |\CommandIndex{makeindex}| <file>.nlo -s nomencl.ist -o <file>.nls|
1131:  \end{example}
1132:
1133:  \section{Dokumentenende}
1134:
1135:  Jede wissenschaftliche Arbeit endet\index{Dokumentenende} mit
1136:  einer Reihe von Verzeichnissen, die dem Leser als zusätzlich
1137:  Informationsquelle dienen. Die Bereitstellung dieser Verzeichnisse
1138:  ist Aufgabe von \fomdes, welche in der Datei \FileIndex{\fomdessty}
1139:  implementiert ist.
1140:
1141:  \DescMacro{\usepackage}
1142:  Die Verwendung des Stils \fomdes\ erfolgt analog der Verwendung aller
1143:  anderen \LaTeX"~Stildateien. Somit wird in der Präambel der Hauptdatei
1144:  eines neuen Dokuments die Zeile
1145:  {\small|\usepackage|\oarg{Optionen}|{fomdes}|} eingetragen.
1146:
1147:  Die Angabe mehrerer Parameter im Optionsteil des Kommandos
1148:  {\small|\usepackage|} ist möglich, wobei einzelne Optionen durch
1149:  ein Komma voneinander getrennt werden.
1150:
1151:  \subsection{Optionen}
1152:
1153:  Zur Wahl der Verzeichnisse am Ende einer wissenschaftlichen Arbeit
1154:  verfügt \fomdes\ über die im Folgenden gezeigten Optionen. Bei
1155:  unbekannten oder nicht unterstützten Optionen wird eine Warnmeldung
1156:  ausgegeben, die zur Information des Benutzers dient.
1157:
1158:  \DescOption{listings} Diese Option aktiviert die Verwendung des
```

```
1159:  Verzeichnisses für Codelistings\index{Gleitobjekt>Codelisting}. Der
1160:  Standardwert für die Verwendung des Verzeichnisses für Codelistings
1161:  ist \defval{aus}.
1162:
1163:  \DescOption{glossary} Diese Option aktiviert die Verwendung des
1164:  Glossars\index{Glossar} beziehungsweise des
1165:  Sachwortverzeichnisses\index{Sachwortverzeichnis\See{Glossar}}.
1166:  Der Standardwert für die Verwendung des Glossars ist \defval{aus}.
1167:  Wird diese Option verwendet, dann wird während des Kompilierens eine
1168:  Meldung ausgegeben, die den Benutzer darauf hinweist, dass zur
1169:  Erstellung des Glossars das Kommando
1170:  \CommandIndex{bibtex}| <file>.gls| aufgerufen werden muss.
1171:
1172:  \DescOption{glossrefpage} Diese Option legt fest, ob im
1173:  Sachwortverzeichnis eine Seitenreferenz für das jeweils erste
1174:  Auftreten eines Glossareintrages enthalten sein soll. Sie ist aber
1175:  nur dann wirksam, wenn \fomdes\ ebenfalls mit der Option
1176:  \option{glossary} geladen wird. Anderenfalls hat die Option
1177:  \option{glossrefpage} keine Auswirkungen. Als Standardwert wird
1178:  \defval{keine Seitenreferenz} im Glossar verwendet.
1179:
1180:  \DescOption{nobibliography} Diese Option verhindert die Aktivierung
1181:  des Literaturverzeichnisses\index{Literaturverzeichnis}. Der
1182:  Standardwert für die Verwendung des Literaturverzeichnisses ist
1183:  \defval{an}. Wird diese Option \emph{nicht} verwendet, dann wird
1184:  während des Kompilierens eine Meldung ausgegeben, die den Benutzer
1185:  darauf hinweist, dass zur Erstellung des Literaturverzeichnisses das
1186:  Kommando \CommandIndex{bibtex}| <file>.aux| aufgerufen werden muss.
1187:
1188:  \DescOption{index} Diese Option aktiviert die Verwendung des
1189:  Indexes\index{Index} beziehungsweise des
1190:  Stichwortverzeichnisses\index{Stichwortverzeichnis\See{Index}}.
1191:  Der Standardwert für die Indexerzeugung ist \defval{aus}. Wird diese
1192:  Option verwendet, dann wird während des Kompilierens eine Meldung
1193:  ausgegeben, die den Benutzer darauf hinweist, dass zur Erstellung des
1194:  Indexes das Kommando \CommandIndex{makeindex}| <file>.idx| aufgerufen
1195:  werden muss.
1196:
1197:  \DescOption{nopagebreak} Diese Option verhindert den erzwungenen
1198:  Seitenwechsel nach jeweils jedem der oben beschriebenen Verzeichnisse.
1199:  Sie kann besonders bei kurzen Verzeichnissen nützlich sein. Der
1200:  Standardwert ist \defval{Seitenumbruch}.
1201:
1202:  \subsection{Makros}
1203:
1204:  \DescMacro{\PrintListings}
1205:  Das Makro {\small|\PrintListings|\oarg{Überschrift}} dient der
1206:  Erstellung des Verzeichnisses für Codelistings. Hierbei ist es durch
1207:  den optionalen Parameter möglich, die Überschrift des
1208:  Listingverzeichnisses zu ändern. Der Standardwert für die Überschrift
1209:  des Listingverzeichnisses ist \defval{Listings}.
```

```
1210:
1211:  Zur Aktivierung des Makros \MacroIndex{\PrintListings} ist es
1212:  notwendig, den Stil \fomdes\ zuvor mit der Option
1213:  \OptionIndex{listings} zu laden. Anderenfalls bleibt die Verwendung
1214:  von \MacroIndex{\PrintListings} wirkungslos.
1215:
1216:  \DescMacro{\PrintGlossary}
1217:  Das Makro {\small|\PrintGlossary| \oarg{Üb\-er\-schrift}
1218:  \optionex{all} \marg{Glos\-sar\-da\-tei}} dient der Erstellung des
1219:  Glossars oder auch Sachwortverzeichnisses. Dieses Makro verfügt, im
1220:  Gegensatz zu dem zuvor beschriebenen Makro, nicht nur über optionale
1221:  Parameter, sondern auch über einen Pflichtparameter!
1222:
1223:  Mit Hilfe der Option \oarg{Überschrift} kann die Überschrift des
1224:  Glossars individuell verändert werden. Der Standardwert für die
1225:  Glossarüberschrift ist \defval{Glossar}.
1226:
1227:  Der zweite optionale Parameter \optionex{all} erzwingt die Ausgabe
1228:  aller in der Bibliothek vorhandenen Glossareinträge. Dieser Parameter
1229:  ist besonders dann hilfreich, wenn in der Glossarbibliothek
1230:  zusätzliche Datensätze vorhanden sind, die zuvor im Text noch nicht
1231:  verwendet wurden.
1232:
1233:  \DescOnlyNoIdx{Achtung!\index{Glossar>Achtung}} Die Angabe des
1234:  optionalen Parameters \optionex{all} führt zur Warnmeldung
1235:  \option{Package gloss Warning: Term '*' on page \holder{n} not
1236:  defined in glossary 'default' on input line \holder{m}}!
1237:
1238:  \DescOnlyNoIdx{Empfehlung\index{Glossar>Empfehlung}} Von der
1239:  Verwendung des Parameters \optionex{all} sollte \emph{nur} zu
1240:  Testzwecken und in Ausnahmefällen Gebrauch gemacht werden, da
1241:  Sachwortverzeichnisse in wissenschaftlichen Arbeiten ausschließlich
1242:  Begriffe enthalten, die auch im Text verwendet werden!
1243:
1244:  Auf Grund der Tatsache, dass \fomdes\ zur Erzeugung eines Glossars
1245:  das Packet \DependsIndex{gloss} verwendet, müssen die einzelnen
1246:  Glossareinträge in einer externen Datei gespeichert werden. Die
1247:  Angabe dieses Dateinamens ist Aufgabe des Pflichtparameters
1248:  \marg{Glossardatei}. Diese Angabe erfogt jedoch immer \emph{ohne}
1249:  die Dateiendung!
1250:
1251:  Da die Erstellung eines Glossars etwas komplexer ist, soll das
1252:  folgende Beispiel\index{Beispiel>Glossar} der Darstellung aller
1253:  notwendigen Schritte zur Erstellung des Glossars dienen.
1254:
1255:  \begin{itemize}
1256:  \item Als erstes wird unter dem Dateinamen \file{glsbsp.bib} die
1257:  Bibliothek des Glossars angelegt und in dieser Datei der Datensatzes
1258:  "'gls1"' wie folgt eingetragen:
1259:
1260:  \begin{verbatim}
```

```
1261:   @GLOSSDEF{gls1,
1262:     word       = {Lange Beschreibung},
1263:     short      = {kurz},
1264:     definition = {Hier ist beschrieben,
1265:                   was "‘kurz"’ bedeutet.}}
1266:  \end{verbatim}
1267:
1268:  \item Im nächsten Schritt wird ein \LaTeX"~Dokument mit dem Dateinamen
1269:  \file{glsbsp.tex} erzeugt, welches über den folgenden Inhalt verfügt:
1270:
1271:  \begin{verbatim}
1272:   \documentclass{fomsdt}
1273:   \usepackage[glossary,nopagebreak]{fomdes}
1274:   \begin{document}
1275:   In diesem Text befindet sich
1276:   "‘\gloss[short]{gls1}"’ ein Hinweis
1277:   auf {\bf\gloss{gls1}}.
1278:   % glsbsp.bib OHNE .bib einbinden!
1279:   \PrintGlossary{glsbsp}
1280:   \end{document}
1281:  \end{verbatim}
1282:
1283:  \item Nachdem zuvor alle Dateien gespeichert wurden, kann mit dem
1284:  Kompiliervorgang begonnen werden. Hierfür sind die folgenden Kommandos
1285:  der Reihe nach auszuführen:
1286:
1287:  \begin{verbatim}
1288:   $ pdflatex glsbsp.tex
1289:   $ bibtex glsbsp.gls
1290:   $ pdflatex glsbsp.tex
1291:   $ pdflatex glsbsp.tex
1292:  \end{verbatim}
1293:
1294:  \end{itemize}
1295:
1296:  \DescOnlyNoIdx{Hinweis\index{Glossar>Hinweis}} Werden in der
1297:  Glossarbibliothek Datensätze hinzugefügt oder geändert, ist ein
1298:  erneuter Aufruf des Programms \CommandIndex{bibtex} erforderlich!
1299:
1300:  Unter Umständen kann es sinnvoll sein, die im Glossar verwendete
1301:  Standardseitenreferenz zu ändern. Soll nun beispielsweise die
1302:  voreingestellte Seitenreferenz \defval{(Seite \holder{Seite})}
1303:  in zum Beispiel \defval{(see page \holder{Seite})} geändert werden,
1304:  dann kann dies durch die Anwendung des folgenden Kommandos erfolgen:
1305:
1306:  \begin{example}
1307:  | \renewcommand{\glosspage}[1]{\space(see page~#1)}|
1308:  \end{example}
1309:
1310:  An dieser Stelle noch die vom Paket \DependsIndex{gloss} unterstützten
1311:  Schlüsselwörter zur Definition von Datensätzen in einer
```

```
1312:  Glossarbibliothek.
1313:
1314:  \begin{verbatim}
1315:   @GLOSSDEF{label,              % Pflichtangabe
1316:     word       = "word",        % Pflichtangabe
1317:     definition = "definition", % Pflichtangabe
1318:     short      = "short",       % Optional
1319:     sort-word  = "sort-word",  % Optional
1320:     group      = "group",       % Optional
1321:     heading    = "heading"      % Optional
1322:   }
1323:  \end{verbatim}
1324:
1325:  \DescMacro{\PrintBibliography} Das Makro
1326:  {\small|\PrintBibliography|\oarg{Überschrift}\oarg{Stil}\marg{Litera\-turdatei}}
1327:  dient der Erstellung des Literaturverzeichnisses. Es verfügt ebenfalls
1328:  über zwei optionale Parameter und auch über einen Pflichtparameter!
1329:
1330:  Mit Hilfe der Option \oarg{Überschrift} kann die Überschrift
1331:  des Literaturverzeichnisses individuell verändert werden. Der
1332:  Standardwert für die Überschrift des Literaturverzeichnisses ist
1333:  \defval{Literatur}.
1334:
1335:  Der zweite optionale Parameter \oarg{Stil} ermöglicht die Änderung
1336:  des Layouts des zu erstellenden Literaturverzeichnisses. Als Angabe
1337:  wird hier der Name einer Datei mit der Endung \file{.bst} erwartet!
1338:  Der Standardwert für das Layout des Literaturverzeichnisses ist das
1339:  DIN"~gerechte Layout \defval{dinat}.
1340:
1341:  \DescOnlyNoIdx{Empfehlung\index{Literaturverzeichnis>Empfehlung}} Für
1342:  die Erstellung wissenschaftlicher Arbeiten wird von der Verwendung der
1343:  Option \oarg{Stil} abgeraten, da dies ein Abweichen vom DIN"~gerechten
1344:  Standardlayout zur Folge hat!
1345:
1346:  Auf Grund der Tatsache, dass \fomdes\ zur Erzeugung des
1347:  Literaturverzeichnisses das Packet \DependsIndex{natbib} verwendet,
1348:  müssen die einzelnen Literaturverweise in einer externen Datei
1349:  gespeichert werden. Die Angabe dieses Dateinamens ist Aufgabe des
1350:  Pflichtparameters \marg{Literaturdatei}. Diese Angabe erfogt jedoch
1351:  immer \emph{ohne} die Dateiendung!
1352:
1353:  Da die Erstellung eines Literaturverzeichnisses etwas komplexer ist,
1354:  soll das folgende Beispiel\index{Beispiel>Literaturverzeichnis} der
1355:  Darstellung aller notwendigen Schritte zur Erstellung des
1356:  Literaturverzeichnisses dienen.
1357:
1358:  \begin{itemize}
1359:  \item Als erstes wird unter dem Dateinamen \file{litbsp.bib} die
1360:  Literaturbibliothek angelegt und in dieser Datei der Datensatzes
1361:  "‘bib1"’ wie folgt eingetragen:
1362:
```

```
1363:   \begin{verbatim}
1364:    @BOOKLET{bib1,
1365:      title   = {The Art of Computer Programming},
1366:      author  = {Donald~E. Knuth},
1367:      year    = {1999},
1368:      month   = {Januar},
1369:      url     = {http://www.amazon.de/},
1370:      urldate = {2007-08-01}}
1371:   \end{verbatim}
1372:
1373:   Der hier gezeigten Datensatz dient ausschließlich zur Demonstration!
1374:   Aus diesem Grund wird an dieser Stelle auf die umfangreich vorhandene
1375:   Literatur zur Erstellung von Literaturverzeichnissen verwiesen, da
1376:   eine ausführliche Beschreibung der vielfältigen
1377:   Variationsmöglichkeiten den Rahmen dieser Anleitung sprengen würde.
1378:
1379:   \item Im nächsten Schritt wird ein \LaTeX"~Dokument mit dem Dateinamen
1380:   \file{litbsp.tex} erzeugt, welches über den folgenden Inhalt verfügt:
1381:
1382:   \begin{verbatim}
1383:    \documentclass{fomsdt}
1384:    \usepackage[nopagebreak]{fomdes}
1385:    \begin{document}
1386:    In diesem Text befindet sich ein Verweis
1387:    auf eine Literaturquelle~\cite{bib1}.
1388:    % litbsp.bib OHNE .bib einbinden!
1389:    \PrintBibliography{litbsp}
1390:    \end{document}
1391:   \end{verbatim}
1392:
1393:   \item Nachdem zuvor alle Dateien gespeichert wurden, kann mit dem
1394:   Kompiliervorgang begonnen werden. Hierfür sind die folgenden Kommandos
1395:   der Reihe nach auszuführen:
1396:
1397:   \begin{verbatim}
1398:    $ pdflatex litbsp.tex
1399:    $ bibtex litbsp.aux
1400:    $ pdflatex litbsp.tex
1401:    $ pdflatex litbsp.tex
1402:   \end{verbatim}
1403:
1404:   Zu bemerken bleibt noch, dass einige Versionen des Programms
1405:   \CommandIndex{bibtex} die Angabe der Dateiendung \file{.aux}
1406:   erlauben und andere Implementierungen die Annahme dieser
1407:   Dateiendung verweigern. In diesem Fall würde dann das
1408:   Kommandos \CommandIndex{bibtex}| <file>.aux| zu einer
1409:   Fehlermeldung führen, die in etwa \option{I couldn't open file
1410:   name 'litbsp.aux.aux'} lautet. In einem solchen Fall sollte dann
1411:   die Dateiendung einfach weggelassen werden.
1412:
1413:   \end{itemize}
```

```
1414:
1415: \DescOnlyNoIdx{Hinweis\index{Literaturverzeichnis>Hinweis}} Werden
1416: in der Literaturbibliothek Datensätze hinzugefügt oder geändert, ist
1417: ein erneuter Aufruf des Programms \CommandIndex{bibtex} erforderlich!
1418:
1419: \DescMacro{\PrintIndex}
1420: Das Makro {\small|\PrintIndex|\oarg{Überschrift}\oarg{Spalten}} dient
1421: der Erstellung des Indexes. Hierbei ist es möglich, über die beiden
1422: optionalen Parameter, das Aussehen des Stichwortverzeichnisses zu
1423: beeinflussen.
1424:
1425: Mit Hilfe der Option \oarg{Überschrift} kann die Überschrift des
1426: Indexes individuell verändert werden. Der Standardwert für die
1427: Indexüberschrift ist \defval{Index}.
1428:
1429: Mit Hilfe der Option \oarg{Spalten} ist es möglich, die Anzahl der
1430: Spalten des Stichwortverzeichnisses zu variieren. Gültige Werte für
1431: die Spaltenzahl sind:
1432:
1433: \OptExplain[15mm]{\option{single}}{Wird
1434: {\small|\bs{}PrintIndex[...][single]|} verwendet, dann besteht das
1435: Stichwortverzeichnisses aus einer einzelnen Spalte.}
1436:
1437: \OptExplain[15mm]{\option{double}}{Wird
1438: {\small|\bs{}PrintIndex[...][double]|} verwendet, dann besteht das
1439: Stichwortverzeichnisses aus zwei Spalten.}
1440:
1441: \OptExplain[15mm]{\option{triple}}{Wird
1442: {\small|\bs{}PrintIndex[...][triple]|} verwendet, dann besteht das
1443: Stichwortverzeichnisses aus drei Spalten.}
1444:
1445: Wird der optionale Parameter \oarg{Spalten} nicht angegeben, dann
1446: wird \defval{double} als Standardwert für die Anzahl der Spalten des
1447: Stichwortverzeichnisses verwendet.
1448:
1449: Das folgende kleine Beispiel\index{Beispiel>Index} demonstriert die
1450: Verwendung des Makros \MacroIndex{\PrintIndex} in Verbindung mit der
1451: zur Aktivierung notwendigen Paketoption \OptionIndex{index}.
1452:
1453: \begin{verbatim}
1454:  \documentclass{fomsdt}
1455:  \usepackage[index]{fomdes}
1456:  \begin{document}
1457:  Dieser Text\index{Text} verwendet einen
1458:  Index\index{Index\see{Text}} mit zwei
1459:  Einträgen sowie einem Querverweis im
1460:  Index selbst.
1461:  \PrintIndex
1462:  \end{document}
1463: \end{verbatim}
1464:
```

```
1465:  \DescOnlyNoIdx{Besonderheit\index{Index>Besonderheit}} Im Zusammenhang
1466:  mit der Erstellung eines Stichwortverzeichnisses sind es noch einige
1467:  Besonderheiten zu beachten. Grundsätzlich gilt hierbei, dass die
1468:  alleinige Aktivierung des Indexes, zum Beispiel durch die Verwendung
1469:  von \MacroIndex{\PrintIndex}, nicht ausreicht, um den Index auch
1470:  tatsächlich in das erzeugte Dokument zu übertragen. Hierfür ist
1471:  zusätzlich der Aufruf des Programms \CommandIndex{makeindex}
1472:  erforderlich!
1473:
1474:  Weiterhin ist die Einhaltung eine bestimmte
1475:  Reihenfolge\index{Index>Reihenfolge} der Schritte zur
1476:  Dokumentenerstellung notwendig. Diese Reihenfolge soll nun näher
1477:  erläutert werden:
1478:
1479:  \begin{itemize}
1480:  \item Das Dokument wird zuerst einmal kompiliert. Dies kann durch das
1481:    Kommando \CommandIndex{pdflatex}{\small| <file>.tex|} erfolgen.
1482:  \item Nun \emph{muss} das Dokument ein zweites mal kompiliert werden!
1483:    Somit erfolgt eine erneute Anwendung des Kommandos
1484:    \CommandIndex{pdflatex}{\small| <file>.tex|}. Dieser zweite
1485:    Kompiliervorgang ist erforderlich, da erst hiernach alle
1486:    Querverweise auf die "‘richtigen"’ Seiten verweisen!
1487:  \item Erst jetzt kann das Stichwortverzeichnis richtig erstellt
1488:    werden. Dies geschieht durch Anwendung des Kommandos
1489:    \CommandIndex{makeindex}{\small| <file>.idx|}.
1490:  \item Abschließend wird das Dokument ein weiteres mal mit dem Befehl
1491:    \CommandIndex{pdflatex}{\small| <file>.tex|} kompiliert.
1492:  \end{itemize}
1493:
1494:  \DescOnlyNoIdx{Empfehlung\index{Index>Empfehlung}} Da die
1495:  Indexerstellung mit der zuvor gezeigten Methode einige Probleme, wie
1496:  beispielsweise die fehlerhafte Anzeige deutscher Umlaute, verursacht,
1497:  wird anstelle des Kommandos
1498:  \CommandIndex{makeindex}{\small| <hauptdatei>.idx|} die Erzeugung des
1499:  Stichwortverzeichnisses wie folgt empfohlen\index{Index>Empfehlung}:
1500:
1501:  \begin{example}
1502:  | $ |\CommandIndex{makeindex}| <file>.idx -g -s |\FileIndex{\fomidxist}
1503:  \end{example}
1504:
1505:  \DescMacro{\SeeAlso}
1506:  Das Makro {\small|\SeeAlso|\marg{Verweis}} ist für die einheitliche
1507:  Formatierung von Querverweisen innerhalb des Indexes oder auch für
1508:  Glossareinträge sehr hilfreich. Es erzeugt einen kleinen Pfeil mit
1509:  dahinter befindlichen schräg gestelltem Text. Diese Art der
1510:  Formatierung dient der besseren Hervorhebung.
1511:
1512:  Mit Hilfe der Option \marg{Verweis} kann nicht nur reiner Text
1513:  verarbeitet werden, vielmehr besteht auch die Möglichkeit, weitere
1514:  Makros mit einzubinden, wie folgende kleine Beispiele demonstrieren.
1515:
```

```
1516:  Das erste Beispiel\index{Beispiel>Index} veranschaulicht die
1517:  Verwendung des Makros \MacroIndex{\SeeAlso} im Zusammenhang mit
1518:  Querverweisen innerhalb eines Indexes. Nimmt man zum Beispiel an,
1519:  dass im Index des Dokuments bereits ein Eintrag für "‘Bildschirm"’
1520:  existiert und möchte man nun den Eintrag "‘Monitor"’ mit einem
1521:  Querverweis auf "‘Bildschirm"’ zum Index hinzufügen, dann würde man
1522:  im Quelltext des Dokuments folgendes eingeben:
1523:
1524:  \begin{verbatim}
1525:   ... Dieser Monitor\index{Monitor\SeeAlso{Bildschirm}}
1526:   zeigt die Zustandsparameter des Gesamtsystems, ...
1527:  \end{verbatim}
1528:
1529:  Nach erneuter Erstellung des Ausgabedokuments stellt sich das
1530:  Ergebnis im Index wie folgt dar:
1531:
1532:  \begin{example}
1533:  Monitor $\to$ \textsl{Bildschirm} \dots\dots \holder{Seite}
1534:  \end{example}
1535:
1536:  Das zweite Beispiel\index{Beispiel>Glossar} zeigt die Verwendung
1537:  des Makros \MacroIndex{\SeeAlso} zusammen mit den Einträgen eines
1538:  Glossars. Hierfür wird angenommen, dass in der Glossardatei
1539:  \file{glsbsp.bib} die beiden Datensätze "‘pin"’ und "‘pinpad"’
1540:  bereits eingefügt wurden. Nun wird unter Anwendung des Makros
1541:  \MacroIndex{\SeeAlso} im Feld \option{definition} des Datensatzes
1542:  \option{pinpad} der Querverweis auf den Datensatz \option{pin} wie
1543:  hier dargestellt eingefügt:
1544:
1545:  \begin{verbatim}
1546:   @GLOSSDEF{pinpad,
1547:     word       = {PIN"~Pad},
1548:     definition = {Das PIN"~Pad wird zur Eingabe
1549:                   der \SeeAlso{\gloss[short]{pin}}
1550:                   verwendet.}}
1551:  \end{verbatim}
1552:
1553:  Nach erneuter Erstellung des Ausgabedokuments stellt sich das
1554:  Ergebnis im Glossar wie folgt dar:
1555:
1556:  {\small\paragraph{\sf PIN"~Pad}Das PIN"~Pad wird zur Eingabe der
1557:  $\to$ \textsl{PIN} verwendet.}
1558:
1559:  \section{Absatzdefinition}
1560:
1561:  Die spezielle Formatierung von Absätzen\index{Absatzformatierung}
1562:  ist in \LaTeX\ eine nicht"~triviale Angelegenheit. Jedoch ist es
1563:  in wissenschaftlichen Arbeiten üblich, Textpassagen besonders
1564:  hervorzuheben. Daher bietet die Absatzdefinition \fompar, die in
1565:  der Datei \FileIndex{\fomparsty} implementiert ist, dem Benutzer
1566:  einige hilfreiche Makros zur Hervorhebung von Absätzen.
```

```
1567:
1568: \DescMacro{\usepackage}
1569: Die Verwendung des Stils \fompar\ erfolgt analog der Verwendung aller
1570: anderen \LaTeX"~Stildateien. Somit wird in der Präambel der Hauptdatei
1571: die Zeile {\small|\usepackage{fompar}|} eingefügt, wobei die Angabe
1572: von Optionen nicht erforderlich ist.
1573:
1574: \subsection{Makros und Umgebungen}
1575:
1576: Nachdem der Stil \fompar\ geladen wurde, stehen dem Anwender die hier
1577: beschriebenen Makros zur Verfürgung.
1578:
1579: \DescEnv{Citation} Längere Zitate werden in wissenschaftlichen
1580: Arbeiten in der Regel in eigenen, zusammenhängenden Absätzen
1581: dargestellt. Dabei werden diese Zitate nicht nur durch doppelte
1582: Anführungszeichen hervorgehoben, sondern oft auch mit einem
1583: zusätzlichen linken und rechten Einzug versehen.
1584:
1585: Die Formatierung dieser Art Textpassagen ist Aufgabe der Umgebung
1586: {\small|\begin{Citation} ... \end{Citation}|}.
1587:
1588: Neben der optionalen Angabe des \oarg{Abstands} zum vorherigen
1589: und nachfolgenden Text, müssen zur Verwendung dieser Umgebung der
1590: \marg{linke} und \marg{rechte} Absatzeinzug unbedingt angegeben
1591: werden. Die genaue Anwendung der Umgebung {\small|Citation|} zeigt
1592: folgendes Beispiel\index{Beispiel>Absatz}.
1593:
1594: \begin{verbatim}
1595:  \begin{Citation}  % Begin of the environment
1596:        [2\parskip] % Leading/following gap (optional)
1597:        {12mm}      % Left indent
1598:        {1.5cm}     % Right indent
1599:  Text des Zitats...
1600:  \end{Citation}    % End of the environment
1601: \end{verbatim}
1602:
1603: Der Standardwert des optionalen Arguments \oarg{Abstand} ist auf
1604: den aktuellen Wert von \defval{\bs{}parskip} eingestellt.
1605:
1606: \DescMacro{\Remark}
1607: Das Makro
1608: {\small|\Remark|\oarg{Einzug}\oarg{Abstand}\marg{Label}\marg{Text}}
1609: erzeugt einen linksseitig eingezogenen Absatz, dem ein linksbündig
1610: ausgerichteter Bezeichner vorangestellt ist. Daher eignet sich dieses
1611: Makro gut für die Hervorhebung wichtiger Informationen innerhalb eines
1612: Textes.
1613:
1614: Die optionalen Parameter \oarg{Einzug} und \oarg{Abstand} ermöglichen
1615: sowohl die Anpassung des linksseitigen Einzugs als auch die Änderung
1616: des Abstandes zum vorherigen und nachfolgenden Text. Als Standardwert
1617: des linken Einzugs werden \defval{3cm} und für den Abstand zum
```

```
1618:   umgebenden Text wird der aktuell eingestellte Wert von
1619:   \defval{\bs{}parskip} verwendet.
1620:
1621:   An dieser Stelle  nun ein kleines Beispiel\index{Beispiel>Absatz} zur
1622:   Illustration der Verwendung des Makros {\small|\Remark|}.
1623:
1624:   \begin{verbatim}
1625:    \Remark[2cm]       % left indent
1626:           [2\parskip] % gap before/after
1627:           {\bf Label} % label, bold
1628:           {Text...}   % body text
1629:   \end{verbatim}
1630:
1631:   \DescMacro{\Item} Das Makro
1632:   {\small|\Item|\oarg{Einzug}\oarg{Abstand}\marg{Label}\marg{Text}}
1633:   erzeugt, genau wie das Makro {\small|\Remark|}, einen linksseitig
1634:   eingezogenen Absatz. Jedoch wird hier dem Absatz ein rechtsbündiger
1635:   Bezeichner vorangestellt. Somit eignet sich dieses Makro gut für die
1636:   Hervorhebung von zum Beispiel Befehlen oder Kommandos eines Programms.
1637:
1638:   Der Standardwert des Parameters \oarg{Einzug} verwendet \defval{4cm}
1639:   und der Parameter \oarg{Abstand} ist auf den aktuellen Wert von
1640:   \defval{\bs{}parskip} eingestellt. Ansonsten erfolgt die Verwendung
1641:   des Makros {\small|\Item|} analog zur Verwendung des Makros
1642:   {\small|\Remark|}.
1643:
1644:   Werden innerhalb einer wissenschaftlichen Arbeit beide zuvor
1645:   beschriebenen Makros {\small|\Remark|} und {\small|\Item|} häufiger
1646:   verwendet und weichen dabei der Einzug und/oder der Textabstand von
1647:   den Standardwerten ab, bietet sich eine grundsätzliche Anpassung der
1648:   Standardwerten an. Zu diesem Zweck werden von \fompar\ die im
1649:   Folgenden beschriebenen Makros angeboten.
1650:
1651:   \DescMacro{\RemarkIndent} Das Makro
1652:   {\small|\RemarkIndent|\marg{Einzug}} ändert den Standardwert des
1653:   linksseitigen Einzugs des Makros {\small|\Remark|} auf den im Argument
1654:   \oarg{Einzug} angegebenen Wert.
1655:
1656:   \DescMacro{\DefaultRemarkIndent} Das Makro
1657:   {\small|\DefaultRemarkIndent|} setzt den Standardwert des
1658:   linksseitigen Einzugs des Makros {\small|\Remark|} auf den Standard
1659:   von \defval{3cm} zurück.
1660:
1661:   \DescMacro{\RemarkOffset} Das Makro
1662:   {\small|\RemarkOffset|\marg{Abstand}} ändert den Standardwert des
1663:   Abstandes zum umgebenden Text des Makro {\small|\Remark|} auf den
1664:   im Argument \oarg{Abstand} angegebenen Wert.
1665:
1666:   \DescMacro{\DefaultRemarkOffset} Das Makro
1667:   {\small|\DefaultRemarkOffset|} setzt den Standardwert des Abstandes
1668:   zum umgebenden Text des Makros {\small|\Remark|} auf den Standard
```

```
1669:   \defval{\bs{}parskip} zurück.
1670:
1671:   \DescMacro{\ItemIndent} Das Makro {\small|\ItemIndent|\marg{Einzug}}
1672:   ändert den Standardwert des linksseitigen Einzugs des Makros
1673:   {\small|\Item|} auf den im Argument \oarg{Einzug} angegebenen Wert.
1674:
1675:   \DescMacro{\DefaultItemIndent} Das Makro {\small|\DefaultItemIndent|}
1676:   setzt den Standardwert des linksseitigen Einzugs des Makros
1677:   {\small|\Item|} auf den Standard von \defval{4cm} zurück.
1678:
1679:   \DescMacro{\ItemOffset} Das Makro {\small|\ItemOffset|\marg{Abstand}}
1680:   ändert den Standardwert des Abstandes zum umgebenden Text des Makro
1681:   {\small|\Item|} auf den im Argument \oarg{Abstand} angegebenen Wert.
1682:
1683:   \DescMacro{\DefaultItemOffset} Das Makro {\small|\DefaultItemOffset|}
1684:   setzt den Standardwert des Abstandes zum umgebenden Text des Makros
1685:   {\small|\Item|} auf den Standard \defval{\bs{}parskip} zurück.
1686:
1687:   Zur Verbesserung des Verständnisses der Anwendung der zuvor
1688:   beschriebenen Makros an dieser Stelle noch ein
1689:   Kurzbeispiel\index{Beispiel>Absatz}. Dabei sei nochmals daran
1690:   erinnert, dass die Makros {\small|\Remark|} und {\small|\Item|} sich
1691:   nur in der Ausrichtung ihrer Bezeichner unterscheiden.
1692:
1693:   \begin{verbatim}
1694:    \ItemIndent{25mm}
1695:    \ItemOffset{1.5\parskip}
1696:    \Item{\tt copy()}{Der Befehl kopiert ...}
1697:    \Item{\tt move()}{Der Befehl verschiebt ...}
1698:    \Item{\tt list()}{Der Befehl listet ...}
1699:    \DefaultItemIndent
1700:    \DefaultItemOffset
1701:   \end{verbatim}
1702:
1703:   \DescOnlyNoIdx{Achtung!} Zu Beginn des Dokuments werden von \fompar\
1704:   die Standardwerte der Makros {\small|\Remark|} und {\small|\Item|}
1705:   neu eingestellt. Aus diesem Grund ist eine Verwendung der Makros
1706:   {\small|\RemarkIndent|}, {\small|\RemarkOffset|},
1707:   {\small|\ItemIndent|} und {\small|\ItemOffset|} in der Präambel nicht
1708:   empfehlenswert!
1709:
1710:   \DescMacro{\Description} Das Makro
1711:   {\small|\Description|\oarg{Format}\marg{Überschrift}\marg{Text}}
1712:   generiert einen Absatz, der über eine Überschrift verfügt und ist
1713:   als Alternative zum \LaTeX"~eigenen Makro {\small|\paragraph|}
1714:   gedacht. Dabei unterscheidet sich {\small|\Description|} in zwei
1715:   grundlegenden Eigenschaften vom Makro {\small|\paragraph|}.
1716:
1717:   \begin{itemize}
1718:   \item Der erste Unteschied besteht darin, dass die im Parameter
1719:   \marg{Überschrift} übergebene Überschrift in einer eigenen Zeile
```

```
1720: dargestellt wird, ohne dabei den gesamten Absatz zu "`zerreißen"'.
1721: \item Der zweite Unterschied zum Standardmakro {\small|\paragraph|}
1722: ist dadurch gegeben, dass die im Parameter \marg{Über\-schrift}
1723: übergebene Überschrift \emph{niemals} im Inhaltsverzeichnis der
1724: Ausarbeitung auftauchen wird, wie dies eventuell bei einer Verwendung
1725: von {\small|\paragraph|} der Fall sein könnte.
1726: \end{itemize}
1727:
1728: Neben den Pflichtparametern verfügt das Makro {\small|\Description|}
1729: auch über einen optionalen Parameter mit dessen Hilfe die
1730: Absatzüberschrift formatiert werden kann. Die Standardformatierung
1731: der Überschrift ist \defval{fett}. Weiterhin ist die Angabe folgender
1732: Optionen möglich:
1733:
1734: \OptExplain{\option{b}}{Die Überschrift ist fett (bold) (Standard).}
1735: \OptExplain{\option{n}}{Die Überschrift nicht (not) formatiert.}
1736: \OptExplain{\option{s}}{Die Überschrift ist schräg gestellt (slanted).}
1737: \OptExplain{\option{u}}{Die Überschrift ist unterstrichen (underlined).}
1738:
1739: Die bespielhafte Erzeugung eines Absatzes mit unterstrichener
1740: Überschrift zeigt der folgende kurze
1741: Codeschnipsel\index{Beispiel>Absatz}.
1742:
1743: \begin{verbatim}
1744:  \Description[u]{Überschrift}{Text...}
1745: \end{verbatim}
1746:
1747: \section{Gleitobjektdefinition}
1748:
1749: Die Besonderheiten der Formatierung so genannter
1750: Gleitobjekte\index{Gleitobjekt}, dazu zählen Abbildungen, Tabellen
1751: und im weiteren Sinne auch Codelistings, werden in wissenschaftlichen
1752: Ausarbeitungen oft unterschätzt. Daher bietet die
1753: Gleitobjektdefinition \fomfod, die in der Datei \FileIndex{\fomfodsty}
1754: implementiert ist, entsprechende Makros an, die der einfachen und
1755: einheitlichen Integration von Gleitobjekten in wissenschaftlichen
1756: Arbeiten dienen.
1757:
1758: \DescMacro{\usepackage}
1759: Die Verwendung des Stils \fomfod\ erfolgt analog der Verwendung
1760: aller anderen \LaTeX"~Stildateien. Somit wird die Zeile
1761: {\small|\usepackage|\oarg{Optionen}|{fomfod}|} in der Präambel
1762: der Hauptdatei eingefügt.
1763:
1764: Die Angabe mehrerer Parameter im Optionsteil des Kommandos
1765: {\small|\usepackage|} ist möglich, wobei einzelne Optionen
1766: durch ein Komma voneinander getrennt werden.
1767:
1768: \subsection{Optionen}
1769:
1770: Der Stil \fomfod\ bietet Optionen zur grundsätzlichen Anpassung der
```

```
1771:  Gleitobjekte einer wissenschaftlichen Arbeit an, die im Folgenden
1772:  näher beschrieben sind. Bei unbekannten oder nicht unterstützten
1773:  Optionen wird eine Warnmeldung ausgegeben, die der Information des
1774:  Benutzers dient.
1775:
1776:  \DescOption{classiccaption}
1777:  Diese Option aktiviert die Verwendung der Serifenschrift für
1778:  die Beschriftungen der einzelnen
1779:  Gleitobjekte\index{Gleitobjekt>Beschriftung}. Als Standardeinstellung
1780:  für Objektbeschriftungen wird eine \defval{serifenlose Schriftart}
1781:  verwendet.
1782:
1783:  \DescOption{listingcolored}
1784:  Diese Option aktiviert die Unterstützung der farblichen Hervorhebung
1785:  von Schlüsselwörtern und Kommentaren innerhalb von Codelistings. Der
1786:  Standardwert für die farbliche Hervorhebung ist \defval{nicht farbig}.
1787:
1788:  \subsection{Makros und Umgebungen}
1789:
1790:  Die von \fomfod\ angebotenen Makros zur Einbindung von Gleitobjekten
1791:  lassen sich in die drei Kategorien Abbildungen, Tabellen und
1792:  Codelistings unterteilen, die im Folgenden auch in dieser Reihenfolge
1793:  vorgestellt werden.
1794:
1795:  \DescMacro{\SinglePicture}
1796:  Die Abbildung\index{Abbildung\See{Gleitobjekt}} stellt in
1797:  wissenschaftlichen Arbeiten die wohl am häufigsten verwendete Form
1798:  von Gleitobjekten\index{Gleitobjekt>Abbildung} dar. Daher bietet das
1799:  Makro {\small|\SinglePicture| \oarg{Brei\-te} |[box]|
1800:  \marg{Be\-schrif\-tung}
1801:  \marg{Re\-fe\-renz\-la\-bel} \marg{Bild\-da\-tei}
1802:  \marg{Quel\-len\-an\-ga\-be}}
1803:  einem Benutzer die Möglichkeit, auf einfache Weise die Integration
1804:  von Abbildungen zu realisieren.
1805:
1806:  Der optionale Parameter \oarg{Brei\-te} ermöglicht eine Anpassung
1807:  der Abbildungsbreite in Abbhängikeit von der aktuellen Textbreite
1808:  \MacroIndex{\textwidth}. Die Höhe der Abbildung wird vom
1809:  \LaTeX"~System automatisch angepasst. Der Standardwert für die
1810:  Abbildungsbreite ist \defval{0.9}, also 90\% von
1811:  \option{\bs{}textwidth}.
1812:
1813:  Der optionale Parameter \option{|[box]|} erzeugt einen Rahmen um die
1814:  eingebundene Abbildung. Der Standardwert für Abbildungsrahmen ist
1815:  \defval{aus}.
1816:
1817:  \DescOnlyNoIdx{Empfehlung} Von der Darstellung eines Rahmens um
1818:  Abbildungen sollte nur in seltenen Fällen gebrauche gemacht werden,
1819:  da sich Rahmen und Linien jeglicher Art störend auf den Lesefluss
1820:  auswirken können\index{Gleitobjekt>Abbildung>Empfehlung}.
1821:
```

```
1822: Der Pflichtparameter \marg{Be\-schrif\-tung} gestattet die Angabe der
1823: Abbildungsunterschrift.
1824:
1825: Der Pflichtparameter \marg{Re\-fe\-renz\-la\-bel} ermöglicht die
1826: Definition eines Bezeichners, mit dessen Hilfe auf die entsprechende
1827: Abbildung Bezug genommen werden kann. Üblicherweise folgt das Label
1828: einer Abbildung dem Schema \option{fig:\holder{Label}}.
1829:
1830: Der Pflichtparameter \marg{Bild\-da\-tei} verwendet den Namen
1831: der Abbildungsdatei. Dabei können Pfade zu unterschiedlichen
1832: Abbildungsquellverzeichnissen mit Hilfe des \LaTeX"~eigenen Makros
1833: \MacroIndex{\graphicspath}\marg{re\-la\-ti\-ver Pfad}
1834: gesteuert werden.
1835:
1836: Der Parameter \marg{Quel\-len\-an\-ga\-be} gehört ebenfalls zu den
1837: Pflichtangaben des Makros \option{\bs{}SinglePicture} und stellt im
1838: Hinblick auf wissenschaftliche Ausarbeitungen einen eher wichtigen
1839: Parameter dar.
1840:
1841: \DescOnlyNoIdx{Hinweis} Es ist nicht nur höflich sondern auch zwingend
1842: erforderlich, in wissenschaftlichen Arbeiten das geistige Eigentum
1843: Dritter entsprechend zu Kennzeichnen! Daher wird bei Abbildungen, die
1844: einer anderen als die eigene Geistesleistung entspringen \emph{immer}
1845: die Quelle angegeben. Diese Quellenangabe erfolgt typischerweise
1846: rechtsbündig und direkt unterhalb der jeweiligen
1847: Abbildung\label{ImgSrcAdvice}\protect.
1848:
1849: Zur Illustration und zur Verbesserung des Verständnisses nun ein
1850: Beispiel zur Verwendung des Makros
1851: \MacroIndex{\SinglePicture}\index{Beispiel>Abbildung}.
1852:
1853: \begin{verbatim}
1854:  \SinglePicture[1.2] % 120% of text width
1855:                [box] % Including an image box
1856:                {Die Darstellung von Spiralen}
1857:                {fig:SpiraleA}
1858:                {spirale} % image file name
1859:                {Quelle: www.spiralen.de}
1860: \end{verbatim}
1861:
1862: \DescMacro{\DoublePicture}
1863: In wissenschaftlichen Arbeiten kommt es häufig vor, dass
1864: Abbildungen\index{Gleitobjekt>Abbildung} zum Beispiel zum Vergleich
1865: gegenüber gestellt werden. Hierfür bietet das Makro
1866: {\small|\DoublePicture| |[box]| \marg{Haupt\-be\-schrif\-tung}
1867: \marg{Haupt\-re\-fe\-renz\-la\-bel} \marg{lin\-ke Ab\-bil\-dung}
1868: \marg{rech\-te Ab\-bil\-dung}} den notwendigen
1869: Rahmen.
1870:
1871: Der optionale Parameter \option{|[box]|} erzeugt einen Rahmen um die
1872: jeweilige Unterabbildung. Der Standardwert für Abbildungsrahmen ist
```

```
1873:  \defval{aus}.
1874:
1875:  \DescOnlyNoIdx{Empfehlung} Von der Darstellung eines Rahmens um
1876:  Abbildungen sollte nur in seltenen Fällen gebrauche gemacht werden,
1877:  da sich Rahmen und Linien jeglicher Art störend auf den Lesefluss
1878:  auswirken können\index{Gleitobjekt>Abbildung>Empfehlung}.
1879:
1880:  Der Pflichtparameter \marg{Haupt\-be\-schrif\-tung} gestattet die
1881:  Angabe der Hauptunterschrift der Doppelabbildung.
1882:
1883:  Der Pflichtparameter \marg{Haupt\-re\-fe\-renz\-la\-bel} ermöglicht
1884:  die Definition eines Bezeichners, mit dessen Hilfe auf die
1885:  entsprechende Doppelabbildung Bezug genommen werden kann.
1886:  Üblicherweise folgt das Label einer Doppelabbildung dem Schema
1887:  \option{fig:\holder{Hauptlabel}}.
1888:
1889:  Die beiden Pflichtparameter \marg{lin\-ke Ab\-bil\-dung} und
1890:  \marg{rech\-te Ab\-bil\-dung} dienen der Definition der linken und
1891:  rechten Unterabbildung und sind an dieser Stelle als Platzhalter
1892:  für das im Folgende beschriebene Makro \MacroIndex{\DoubleSubPicture}
1893:  gedacht.
1894:
1895:  \DescMacro{\DoubleSubPicture}
1896:  Die Definition der linken oder rechten Unterabbildung obliegt dem
1897:  Makro {\small|\DoubleSubPicture| \marg{Un\-ter\-be\-schrif\-tung}
1898:  \marg{Un\-ter\-re\-fe\-renz\-la\-bel} \marg{Un\-ter\-bild\-da\-tei}
1899:  \marg{Un\-ter\-quel\-len\-an\-ga\-be}}.
1900:
1901:  Der Pflichtparameter \marg{Un\-ter\-be\-schrif\-tung} gestattet die
1902:  Angabe der Beschriftung der linken beziehungsweise der rechten
1903:  Unterabbildung.
1904:
1905:  Der Pflichtparameter \marg{Un\-ter\-re\-fe\-renz\-la\-bel} ermöglicht
1906:  die vom Hauptreferenzlabel unabhängige Definition eines Bezeichners,
1907:  mit dessen Hilfe auf die entsprechende Unterabbildung Bezug genommen
1908:  werden kann. Üblicherweise folgt das Label einer Abbildung dem Schema
1909:  \option{fig:\holder{Label}}.
1910:
1911:  Der Pflichtparameter \marg{Un\-ter\-bild\-da\-tei} verwendet den
1912:  Namen der linken oder rechten Unterabbildungsdatei. Dabei können
1913:  Pfade zu unterschiedlichen Abbildungsquellverzeichnissen mit Hilfe
1914:  des \LaTeX"~eigenen Makros
1915:  \MacroIndex{\graphicspath}\marg{re\-la\-ti\-ver Pfad} gesteuert
1916:  werden.
1917:
1918:  Der Parameter \marg{Un\-ter\-quel\-len\-an\-ga\-be} gehört ebenfalls
1919:  zu den Pflichtangaben des Makros \option{\bs{}Double\-Sub\-Picture}
1920:  und stellt im Hinblick auf wissenschaftliche Ausarbeitungen einen
1921:  eher wichtigen Parameter dar.
1922:
1923:  An dieser Stelle bitte ebenfalls den Hinweis zur Quellenangabe bei
```

```
1924:   Abbildungen auf Seite~\pageref{ImgSrcAdvice} beachten!
1925:
1926:   \DescOnlyNoIdx{Achtung!} Die individuelle Anpassung der
1927:   Abbildungsbreite ist weder bei {\small|\DoublePicture|} noch
1928:   bei {\small|\DoubleSubPicture|} möglich! Vielmehr werden beide
1929:   Abbildungen so platziert, dass jede in etwa 42\% der aktuellen
1930:   Textbreite verwendet. Zuzüglich einer kleinen Lücke zwischen beiden
1931:   Abbildungen ist somit die Breite der gesamten Doppelabbildung auf
1932:   circa 90\% der verfügbaren Textbreite begrenzt.
1933:
1934:   Zur Illustration und zur Verbesserung des Verständnisses nun ein
1935:   Beispiel zur Verwendung des Makros \MacroIndex{\DoublePicture}
1936:   zusammen mit dem Makro
1937:   \MacroIndex{\DoubleSubPicture}\index{Beispiel>Abbildung}.
1938:
1939:   \begin{verbatim}
1940:    \DoublePicture
1941:         [box] % Including image boxes
1942:         {Der Vergleich von Linien} % Main caption
1943:         {fig:LinienA}              % Main label
1944:         {\DoubleSubPicture         % Left sub picture
1945:               {Gerade Linien}      % Left caption (a)
1946:               {fig:LinienAleft}    % Left label
1947:               {linien}             % Left image file.
1948:               {Quelle: www.linien.de}}
1949:         {\DoubleSubPicture         % Right sub picture
1950:               {Gekrümmte Linien}   % Right caption (b)
1951:               {fig:LinienAright}   % Right label
1952:               {kurven}             % Right image file.
1953:               {Quelle: www.kurven.de}}
1954:   \end{verbatim}
1955:
1956:   \DescOnlyNoIdx{Bemerkung} Manch Einer wird sich fragen: 'Warum nur
1957:   die Aufteilung in zwei voneinander unabhängige Makros?'. Die Antwort
1958:   ist einfach! Das Makro {\small|\DoublePicture|} würde für eine
1959:   vollständige Verarbeitung aller Informationen insgesamt elf Parameter
1960:   benötigen. In \LaTeX\ ist jedoch die maximale Parameteranzahl auf
1961:   insgesamt neun beschränkt. Aus diesem Grund erfolgte eine
1962:   Unterteilung in zwei unabhängige Makros!
1963:
1964:   \DescEnv{PlainFigure}
1965:   Eine weitere Variante zur Einbindung von
1966:   Abbildungen\index{Gleitobjekt>Abbildung} in eine wissenschaftliche
1967:   Arbeit bietet sich durch die Verwendung der frei konfigurierbaren
1968:   Umgebung {\small|\begin{PlainFigure}| \oarg{Brei\-te} |[box]|
1969:   \marg{Be\-schrif\-tung} \marg{Re\-fe\-renz\-la\-bel}
1970:   \marg{Quel\-len\-an\-ga\-be} |... \end{PlainFigure}|}
1971:
1972:   Der optionale Parameter \oarg{Brei\-te} ermöglicht eine Anpassung
1973:   der Abbildungsbreite in Abhängigkeit von der aktuellen Textbreite
1974:   \MacroIndex{\textwidth}. Die Höhe der Abbildung wird vom
```

```
1975:  \LaTeX"~System automatisch angepasst. Der Standardwert für die
1976:  Abbildungsbreite ist \defval{0.9}, also 90\% von
1977:  \option{\bs{}textwidth}.
1978:
1979:  Der optionale Parameter \option{|[box]|} erzeugt einen Rahmen um den
1980:  eingebundenen Inhalt. Der Zwischenraum zwischen Rahmen und Inhalt
1981:  beträgt \defval{5pt}, also fünf Punkte. Dieser Zwischenraum kann von
1982:  außen nicht geändert werden. Der Standardwert für Abbildungsrahmen
1983:  ist \defval{aus}.
1984:
1985:  Der Pflichtparameter \marg{Be\-schrif\-tung} gestattet die Angabe
1986:  der Abbildungsunterschrift.
1987:
1988:  Der Pflichtparameter \marg{Re\-fe\-renz\-la\-bel} ermöglicht die
1989:  Definition eines Bezeichners, mit dessen Hilfe auf die entsprechende
1990:  Abbildung Bezug genommen werden kann. Üblicherweise folgt das Label
1991:  einer Abbildung dem Schema \option{fig:\holder{Label}}.
1992:
1993:  Der Parameter \marg{Quel\-len\-an\-ga\-be} gehört ebenfalls zu den
1994:  Pflichtangaben der Umgebung \option{PlainFigure} und stellt im
1995:  Hinblick auf wissenschaftliche Ausarbeitungen einen eher wichtigen
1996:  Parameter dar.
1997:
1998:  An dieser Stelle bitte ein weiters mal den Hinweis zur Quellenangabe
1999:  bei Abbildungen auf Seite~\pageref{ImgSrcAdvice} beachten!
2000:
2001:  Zur Illustration und zur Verbesserung des Verständnisses nun ein
2002:  Beispiel zur Verwendung der Umgebung
2003:  \EnvIndex{PlainFigure}\index{Beispiel>Abbildung}.
2004:
2005:  \begin{verbatim}
2006:   \begin{PlainFigure}[0.7] % 70% of text width
2007:                      [box] % Including an image box
2008:                      {Darstellung einer Textabbildung}
2009:                      {fig:TextFigure}
2010:                      {Quelle: www.texte.de}
2011:   Hier befindet sich der frei
2012:   wählbare Inhalt der Abbildung...
2013:   \end{PlainFigure}
2014:  \end{verbatim}
2015:
2016:  \DescEnv{Table}
2017:  Tabellen\index{Gleitobjekt>Tabelle}\index{Tabelle\See{Gleitobjekt}}
2018:  bilden ebenfalls einen wichtigen Bestandteil wissenschaftlicher
2019:  Arbeiten. Da es jedoch hin und wieder vorkommt, dass Tabellen nicht
2020:  standardgerecht formatiert werden, bietet \fomfod\ die entsprechend
2021:  konfigurierte Umgebung {\small|\begin{Table}| \marg{Be\-schrif\-tung}
2022:  \marg{Re\-fe\-renz\-la\-bel} \marg{Quel\-len\-an\-ga\-be}
2023:  |... \end{Table}|} an.
2024:
2025:  Diese Umgebung ist in der Lage, die gängigen, zum \LaTeX"~Sys\-tem
```

```
2026:   gehörenden Umgebungen für Tabellendefinitionen zu verarbeiten.
2027:
2028:   Der Pflichtparameter \marg{Be\-schrif\-tung} gestattet die Angabe
2029:   der Überschrift der Tabelle.
2030:
2031:   Der Pflichtparameter \marg{Re\-fe\-renz\-la\-bel} ermöglicht die
2032:   Definition eines Bezeichners, mit dessen Hilfe auf die entsprechende
2033:   Tabelle Bezug genommen werden kann. Üblicherweise folgt das Label
2034:   einer Tabelle dem Schema \option{tab:\holder{Label}}.
2035:
2036:   Der Parameter \marg{Quel\-len\-an\-ga\-be} gehört ebenfalls zu den
2037:   Pflichtangaben der Umgebung \option{Table} und stellt im Hinblick auf
2038:   wissenschaftliche Ausarbeitungen einen eher wichtigen Parameter dar.
2039:
2040:   \DescOnlyNoIdx{Hinweis} Auch bei Tabellen muss, sofern sie
2041:   beziehungsweise deren Inhalt geistiges Eigentum Dritter sind, der
2042:   Ursprung entsprechend kenntlich gemacht werden! Hierbei erfolgt die
2043:   Quellenangabe typischerweise zentriert und direkt unterhalb der
2044:   jeweiligen Tabelle.
2045:
2046:   Aufgrund der vielfältigen Möglichkeiten, eine Tabelle zu gestallten,
2047:   ist die weitere Abstraktion und Vereinfachung der Erstellung von
2048:   Tabellen nur schwer möglich. Daher soll das folgende Beispiel nur
2049:   stellvertretend für die Tabellenerstellung
2050:   sein\index{Beispiel>Tabelle}.
2051:
2052:   \begin{verbatim}
2053:    \begin{Table}
2054:      {Eine kleine Tabelle}     % Table caption
2055:      {tab:small1}              % Reference label
2056:      {Quelle: www.tabellen.de} % Table source
2057:      \begin{tabular}{lcr}      % Table body
2058:      links & mitte & rechts \\
2059:      \end{tabular}
2060:    \end{Table}
2061:   \end{verbatim}
2062:
2063:   \DescOnlyNoIdx{Empfehlung} Zur besseren Kontrolle der Formatierung
2064:   einzelner Tabellenspalte wird die Verwendung der Umgebung
2065:   \EnvIndex{tabularx} des gleichnamigen Pakets \DependsIndex{tabularx}
2066:   empfohlen\index{Gleitobjekt>Tabelle>Empfehlung}.
2067:
2068:   Wird nun eine Tabelle mit der Umgebung \EnvIndex{tabularx} erstellt,
2069:   dann stellt \fomfod\ spezielle Spaltenformatierungsmakros bereit,
2070:   auf die an dieser Stelle näher eingegangen wird.
2071:
2072:   \OptExplain{\option{R}}{Erzeugt eine rechts ausgerichtete
2073:   Tabellenspalte.}
2074:   \OptExplain{\option{C}}{Erzeugt eine zentriert ausgerichtete
2075:   Tabellenspalte.}
2076:   \OptExplain{\option{L}}{Erzeugt eine links ausgerichtete
```

```
2077:   Tabellenspalte.}
2078:   \OptExplain{\option{B}}{Erzeugt eine im Blocksatz ausgerichtete
2079:   Tabellenspalte.}
2080:
2081:   Zur Veranschaulichung an dieser Stelle noch ein kleines Beispiel zur
2082:   Verwendung der \fomfod"~eigenen Umgebung \EnvIndex{Table}, wie sie
2083:   gemeinsam mit der Umgebung \EnvIndex{tabularx} genutzt werden
2084:   kann\index{Beispiel>Tabelle}.
2085:
2086:   \begin{verbatim}
2087:    \begin{Table}
2088:      {Eine weitere Tabelle}
2089:      {tab:tabX}
2090:      {Quelle: www.tabularx.de}
2091:      \begin{tabularx}{8cm}
2092:        {>{\hsize=2cm}L>{\hsize=1cm}C>{\hsize=5cm}R}
2093:      links 2cm & mitte 1cm & rechts 5cm \\
2094:      \end{tabularx}
2095:    \end{Table}
2096:   \end{verbatim}
2097:
2098:   \DescEnv{CodeListing}
2099:   Gerade auf dem Gebiet der Informatik kommt es immer wieder vor,
2100:   dass so genannte Codelistings\index{Gleitobjekt>Codelisting} in
2101:   wissenschaftliche Arbeiten integriert werden müssen. Die Einbindung
2102:   dieser Quelltexte, sowie deren einheitliche Gestaltung, ist die
2103:   vornehmliche Aufgabe der Umgebung {\small|\begin{CodeListing}|
2104:   \oarg{Ein\-zug} \marg{Be\-schrif\-tung} \marg{Re\-fe\-renz\-la\-bel}
2105:   |... \end{CodeListing}|}
2106:
2107:   Der optionale Parameter \oarg{Ein\-zug} kann verwendet werden, um
2108:   den linken und rechten Einzug des Listings zu variieren. Dabei ist
2109:   der linke und rechte Standardeinzug auf eine Breite von \defval{7pt}
2110:   voreingestellt.
2111:
2112:   Der Pflichtparameter \marg{Be\-schrif\-tung} erlaubt die Angabe
2113:   der Überschrift des Listings.
2114:
2115:   Der Pflichtparameter \marg{Re\-fe\-renz\-la\-bel} ermöglicht die
2116:   Verwendung eines Bezeichners, mit dessen Hilfe auf das entsprechende
2117:   Codelisting Bezug genommen werden kann. Üblicherweise folgt das
2118:   Quelltextlabel dem Schema \option{lst:\holder{Label}}.
2119:
2120:   Hier noch ein kleines Anwendungsbeispiel der Codelistingumgebung
2121:   zur Anzeige des Quelltextes des C"~Programms
2122:   "‘Hallo Welt!"’\index{Beispiel>Hallo Welt}, inklusive einer
2123:   farblichen Syntaxhervorhebung.
2124:
2125:   \begin{verbatim}
2126:    \documentclass{fomsdt}
2127:    \usepackage[listingcolored]{fomfod}
```

```
2128:   \begin{document}
2129:   \begin{CodeListing}[35mm]{"‘Hallo Welt!"’}{lst:hello}
2130:     #include <stdio.h>
2131:     void main(void) {
2132:       printf("Hallo Welt!\n");
2133:     }
2134:   \end{CodeListing}
2135:   \end{document}
2136:  \end{verbatim}
2137:
2138:  \DescOnlyNoIdx{Hinweis} Die Umgebung \EnvIndex{CodeListing} ist
2139:  speziell für kurze Listings gedacht, da hierfür der gesamte
2140:  Quellcode zwischen \option{\bs{}begin} und \option{\bs{}end}
2141:  stehen muss. Bei längeren Quelltexten kann dies jedoch zu
2142:  einer nicht zu unterschätzenden Unübersichtlichkeit führen!
2143:
2144:  \DescMacro{\CodeListingFile}
2145:  Im Gegensatz zur zuvor beschriebenen Umgebung für Codelistings
2146:  bietet das Makro {\small|\CodeListingFile|} \oarg{Ein\-zug}
2147:  \marg{Be\-schrif\-tung} \marg{Re\-fe\-renz\-la\-bel}
2148:  \marg{Quell\-code\-da\-tei} die Möglichkeit, den Inhalt ganzer
2149:  Codedateien in eine wissenschaftliche Arbeit einzubinden.
2150:
2151:  Der optionale Parameter \oarg{Ein\-zug} kann verwendet werden, um
2152:  den linken und rechten Einzug des Listings zu variieren. Dabei ist
2153:  linke und rechte Standardeinzug auf eine Breite von \defval{7pt}
2154:  voreingestellt.
2155:
2156:  Der Pflichtparameter \marg{Be\-schrif\-tung} erlaubt die Angabe der
2157:  Überschrift des Listings.
2158:
2159:  Der Pflichtparameter \marg{Re\-fe\-renz\-la\-bel} ermöglicht die
2160:  Verwendung eines Bezeichners, mit dessen Hilfe auf das entsprechende
2161:  Codelisting Bezug genommen werden kann. Üblicherweise folgt das
2162:  Quelltextlabel dem Schema \option{lst:\holder{Label}}.
2163:
2164:  Der Pflichtparameter \marg{Quell\-code\-da\-tei} ermöglicht die
2165:  Angabe des Dateinamens der einzubindenden Quelltextdatei.
2166:
2167:  Hier nochmals das obige Beispiel für das C"~Programm "‘Hallo
2168:  Welt!"’\index{Beispiel>Hallo Welt}, jedoch nun mit der Einbindung
2169:  der entsprechenden Quellcodedatei \file{hello.c}.
2170:
2171:  \begin{verbatim}
2172:   \documentclass{fomsdt}
2173:   \usepackage[listingcolored]{fomfod}
2174:   \begin{document}
2175:   \CodeListingFile{"‘Hallo Welt!"’}{lst:hello}{hello.c}
2176:   \end{document}
2177:  \end{verbatim}
2178:
```

```
2179:  \DescMacro{\ListingFontSize}
2180:  Eventuell kann es vorkommen, dass die Darstellung der Zeichen eines
2181:  Codelistings zu groß oder gar zu klein erscheinen. Dies könnte dann
2182:  unter Umständen zu dem ungewollten Effekt führen, dass einzelnen
2183:  Zeilen des Quelltextes umgebrochen werden. Das könnte durch die
2184:  Anpassung der Schriftgröße behoben werden. Die Einstellung dieser
2185:  Schriftgröße ist Aufgabe des Makros {\small|\ListingFontSize|
2186:  \oarg{Schrift\-grö\-ße}}.
2187:
2188:  Der optionale Parameter \oarg{Schrift\-grö\-ße} erlaubt hierbei
2189:  die Verwendung vordefinierter Größenangaben zur Einstellung der
2190:  bevorzugten Schriftgröße wie folgt:
2191:
2192:  \OptExplain[15mm]{\option{normal}}{Diese Option verwendet
2193:  \option{\bs{}footnotesize} als Größe der Schrift und erlaubt damit
2194:  die Darstellung von etwa 60 Zeichen pro Codezeile.}
2195:  \OptExplain[15mm]{\option{large}}{Diese Option verwendet
2196:  \option{\bs{}small} als Größe der Schrift und erlaubt damit die
2197:  Darstellung von etwa 55 Zeichen pro Codezeile.}
2198:  \OptExplain[15mm]{\option{small}}{Diese Option verwendet
2199:  \option{\bs{}scriptsize} als Größe der Schrift und erlaubt damit
2200:  die Darstellung von etwa 74 Zeichen pro Codezeile.}
2201:
2202:  \DescMacro{\DefaultListingConfig}
2203:  Das Makro {\small|\DefaultListingConfig|} dient der Einstellung
2204:  aller Standardwerte für Codelistings. Einige dieser Einstellungen
2205:  sind von besonderem Interesse und sollen aus diesem Grund etwas
2206:  näher betrachtet werden.
2207:
2208:  \DescOnlyNoIdx{Quelltextsprache}
2209:  Die Quelltextsprache ist nur dann von besonderem Interesse, wenn
2210:  \fomfod\ zuvor mit der Option \option{listingcolored} geladen wurde,
2211:  dass heißt, wenn auch tatsächlich die farbliche Hervorhebung für
2212:  Quellcodes aktiviert wurde. In diesem Fall wird die Syntaxhervorhebung
2213:  für Quelltexte in C++ verwendet!
2214:
2215:  Soll durch ein Listing die Syntaxhervorhebung einer anderen
2216:  Quelltextsprache verwenden, dann kann die Standardeinstellung C++ mit
2217:  Hilfe des Kommandos {\small|\lstset{language=|\holder{Spra\-che}|}|},
2218:  das in \DependsIndex{listings} definiert ist, geändert werden. Alle
2219:  durch \DependsIndex{listings} unterstützten Sprachen, inklusiver
2220:  diverser Dialekte, werden ausführlich in der zu diesem Paket
2221:  gehörenden Dokumentation beschrieben.
2222:
2223:  \DescOnlyNoIdx{Beschriftung}
2224:  Für die Positionierung der Beschriftung eines Codelistings kann keine
2225:  verbindliche Aussage getroffen werden. Sinnvoll erscheint jedoch, die
2226:  Beschriftung an den Anfang des Quelltextes zu setzen, da ein Leser bei
2227:  einer Textunterbrechung durch zum Beispiel ein Codelisting zumindest
2228:  einen Hinweis  darauf erwartet, warum sein Lesefluss gestört wurde.
2229:
```

```
2230:  Sollte man aus irgendeinem Grund von der vorkonfigurierten
2231:  Beschriftungsposition oberhalb des Listings abweichen müssen, dann
2232:  kann dies durch den im Paket \DependsIndex{listings} definierten
2233:  Befehl {\small|\lstset{captionpos=b}|} erreicht werden.
2234:
2235:  \DescOnlyNoIdx{Zeilenumbruch}
2236:  Unter Umständen kann es vorkommen, dass Codezeilen eines eingebundenen
2237:  Listings trotz der Anpassung des Schriftgrades nicht in eine einzelne
2238:  Textzeile passen. Für diesen Fall wurde \fomfod\ so konfiguriert,
2239:  dass das Codelisting selbstständig einen Umbruch der betroffenen
2240:  Zeilen durchführt und eine so umgebrochene Codezeile dann mit \dots\
2241:  beendet und in der nächsten Zeile mit einer Einrückung und weiteren
2242:  \dots\ fortgeführt wird.
2243:
2244:  \section{Vermischtes}
2245:
2246:  Als Sammelbecken für weiter nützliche und hilfreiche Makros dient
2247:  \fomisc, das in der Datei \FileIndex{\fomiscsty} implementiert ist.
2248:  Hierin befinden sich Makros, die von ihrem Sinnzusammenhang in keiner
2249:  anderen Datei des Pakets untergebracht werden können.
2250:
2251:  \DescMacro{\usepackage}
2252:  Die Verwendung des Stils \fomisc\ erfolgt analog der Verwendung aller
2253:  anderen \LaTeX"~Stildateien. Somit wird in der Präambel der Hauptdatei
2254:  die Zeile {\small|\usepackage{fomisc}|} eingefügt, wobei die Angabe
2255:  von Optionen nicht erforderlich ist.
2256:
2257:  \subsection{Makros}
2258:
2259:  \DescMacro{\Raise} Grundsätzlich kann hochgestellter Text durch den
2260:  mathematischen Satz erzeugt werden. Jedoch führt die Verwendung des
2261:  mathematischen Satzes nicht immer zum erwarteten Ergebnis.
2262:  Beispielsweise wird hierbei hochgestellter Text kursiv dargestellt.
2263:  Dies kann durchaus irritierent auf den Leser einer wissenschaftlichen
2264:  Arbeit wirken. Daher wird zur verbesserten Darstellung hochgestellter
2265:  Texte die Verwendung des Makros {\small|\Raise|\marg{Hochtext}}
2266:  empfohlen.
2267:
2268:  \DescMacro{\TodayShort} Manchmal kann es sehr hilfreich sein, das
2269:  aktuelle Datum in einer kurzen Variante anzugeben. Zu diesem Zweck
2270:  wird durch \fomisc\ das Makro {\small|\TodayShort| \oarg{For\-mat}
2271:  \oarg{Trenn\-zei\-chen}} bereitgestellt.
2272:
2273:  Dabei bestimmt der optionale Parameter \oarg{For\-mat} das
2274:  grundsätzlich Aussehen des erzeugten Datums. Als Standardwert für
2275:  diesen Parameters wird \defval{din} verwendet. Weiterhin sind die
2276:  im Folgenden gezeigten vordefinierten Schlüsselwörter für diesen
2277:  Parameter verfügbar:
2278:
2279:  \OptExplain{\option{din}}{Diese Option verwendet \textsc{dd.mm.yyyy}
2280:  zur Formatierung des aktuellen Datums und entspricht damit den
```

```
2281:  Empfehlungen gemäß Norm DIN~5008.}
2282:
2283:  \OptExplain{\option{iso}}{Diese Option verwendet \textsc{yyyy"~mm"~dd}
2284:  zur Formatierung des aktuellen Datums und entspricht damit den
2285:  Empfehlungen gemäß der Normen ISO~8601:2004 und EN~28601.}
2286:
2287:  Die Zeichen, die zur Trennung der einzelnen Bestandteile des Datums
2288:  verwendet werden, sind bereits normengrecht! Sollte dennoch eine
2289:  Anpassung der Trennzeichen erfolgen müssen, zum Beispiel für eine
2290:  US"~amerikanische Formatierung mit Schrägstrichen als Trennzeichen,
2291:  dann kann dieses vom Standard abweichende Trennzeichen durch den
2292:  optionalen Parameter \oarg{Trenn\-zei\-chen} geändert werden. Dieser
2293:  Parameter verwendet \defval{def} als Standardwert.
2294:
2295:  \OptExplain{\option{def}}{Diese Option bewirkt die Verwendung des
2296:  Stan\-dard\-trenn\-zei\-chens für Datumsangaben in Abhängigkeit vom
2297:  gewählten Datumsformat.
2298:
2299:  Im Falle einer ISO"~gerechten Formatierung, also bei Angabe von
2300:  \option{iso} im Parameter \oarg{For\-mat}, wird der Gedankenstrich
2301:  ('-') zur Trennung verwendet.
2302:
2303:  Für den Fall der DIN"~gerechten Formatierung, also bei Angabe von
2304:  \option{din} im Parameter \oarg{For\-mat}, wird ein Punkt ('.') als
2305:  Trennzeichen verwendet.}
2306:
2307:  \OptExplain{\option{\dots}}{Das angegebene, beliebige Zeichens, oder
2308:  auch eine Zeichenkette, wird zur Trennung der Datumsbestandteile
2309:  verwendet. Die Verwendung eines "`leeren"' Zeichens ist ebenfalls
2310:  möglich.}
2311:
2312:  Folgende Übersicht zeigt beispielhaft\index{Beispiel>Kurzdatum} die
2313:  Anwendungsmöglichkeiten des Makros \MacroIndex{\TodayShort} sowie die
2314:  hieraus resultierenden Ergebnisse:
2315:
2316:  \begin{tabular}{lll}
2317:  \option{\bs{}TodayShort}          & 01.08.2007   &
2318:    {$\Rightarrow$\space\scriptsize DIN"~gerecht}\\
2319:  \option{\bs{}TodayShort[din]}     & 01.08.2007   &
2320:    {$\Rightarrow$\space\scriptsize DIN"~gerecht}\\
2321:  \option{\bs{}TodayShort[din][def]}& 01.08.2007   &
2322:    {$\Rightarrow$\space\scriptsize DIN"~gerecht}\\
2323:  \option{\bs{}TodayShort[din][]}   & 01082007     &
2324:    {$\Rightarrow$\space\scriptsize DIN ohne Trennung}\\
2325:  \option{\bs{}TodayShort[din][/]}  & 01/08/2007   &
2326:    {$\Rightarrow$\space\scriptsize US"~amerikanisch}\\
2327:  \option{\bs{}TodayShort[iso]}     & 2007"~08"~01 &
2328:    {$\Rightarrow$\space\scriptsize ISO"~konform}\\
2329:  \option{\bs{}TodayShort[iso][def]}& 2007"~08"~01 &
2330:    {$\Rightarrow$\space\scriptsize ISO"~konform}\\
2331:  \option{\bs{}TodayShort[iso][]}   & 20070801     &
```

```
2332:   {$\Rightarrow$\space\scriptsize ISO ohne Trennung}\\
2333: \option{\bs{}TodayShort[iso][/]}  & 2007/08/01   &
2334:   {$\Rightarrow$\space\scriptsize US"~amerikanisch}\\
2335: \end{tabular}
2336:
2337: Autoren wissenschaftlicher Arbeiten beziehen sich häufig auf Produkte
2338: und Marken von Herstellern, die mit dem Vermerk "‘Registered"’,
2339: "‘Copyright"’ oder "‘Trademark"’ belegt sind. Hierfür werden in der
2340: Regel spezielle Symbole verwendet, die von \fomisc\ bereitgestellt
2341: werden.
2342:
2343: \DescMacro{\Registered} Trägt ein Produkt oder eine Marke den Vermerk
2344: "‘Registered"’, dann kann das Makro {\small|\Registered|} für die
2345: Erzeugung des entsprechenden Symbols verwendet werden.
2346:
2347: \DescMacro{\Copyright} Trägt ein Produkt oder eine Marke den Vermerk
2348: "‘Copyright"’, dann kann das Makro {\small|\Copyright|} für die
2349: Erzeugung des entsprechenden Symbols verwendet werden.
2350:
2351: \DescMacro{\Trademark} Trägt ein Produkt oder eine Marke den Vermerk
2352: "‘Trademark"’, dann kann das Makro {\small|\Trademark|} für die
2353: Erzeugung des entsprechenden Symbols verwendet werden.
2354:
2355: \DescMacro{\Axel} Das Makro {\small|\Axel|} dient der Illustration
2356: der Erzeugung von Texten, deren einzelne Buchstaben "‘aus der Reihe
2357: tanzen"’. Beispiele solcher Texte sind die Eigennamen \TeX\ und
2358: \LaTeX.
2359:
2360: \section*{\Problems}\addcontentsline{toc}{section}{\Problems}
2361:
2362: Dieses Kapitel verschafft einen Überblick über bekannte
2363: Probleme\index{Probleme} und erläutert deren Zusammenhang oder auch
2364: deren Umgehung. Treten während der Verwendung des Pakets \fomsdt\
2365: weitere, bisher nicht bekannte Probleme, Konflikte oder Fehler auf,
2366: dann sollten die Benutzer nicht vor einer entsprechenden Email
2367: zurückschrecken.
2368:
2369: \DescMacro{\PrintIndex}
2370: Dieses durch die Datei \FileIndex{\fomdessty} definiert Kommando
2371: wird ebenfalls durch die Datei \FileIndex{doc.sty}
2372: definiert\index{Probleme>\file{doc.sty}} und ist somit nach der
2373: Installation des Pakets \fomsdt\ doppelt im \LaTeX"~System verfügbar.
2374: Hierdurch sollten jedoch keinerlei Beeinträchtigungen im Umgang mit
2375: dem Paket \fomsdt\ entstehen, da die Datei \file{doc.sty} in aller
2376: Regel nur im Zusammenhang mit der Erstellung der
2377: Quellcodedokumentation von \LaTeX"~Paketen verwendet wird.
2378:
2379: \DescOnlyNoIdx{Leeres Verzeichnis\index{Probleme>Verzeichnis (leer)}}
2380: Die Anwendung der Makros {\small|\Print|\holder{Ver\-zeich\-nis}}
2381: führen unter Umständen zu dem Effekt, dass die korrespondierenden und
2382: automatisch erstellten Verzeichnisse \emph{nur} aus ihrer Überschrift
```

```
2383:   bestehen und somit über keinerlei  sinnvollen Inhalt verfügen.
2384:
2385:   Tritt dieser Effekt auf, dann sollte zunächst einmal überprüft werden,
2386:   ob die benötigten Querverweise überhaupt im Dokument vorhanden sind.
2387:   Als Beispiel hierfür sei das Fehlen von
2388:   {\small|\index{|\holder{Begriff}|}|} bei gleichzeitiger Anwendung
2389:   von {\small|\PrintIndex|} genannt.
2390:
2391:   Als nächstes sollte geprüft werden, ob zur Erstellung des jeweiligen
2392:   Verzeichnisses nicht noch weitere Kompiliervorgänge notwendig sind.
2393:   Gegebenenfalls ist auch eine Anwendung der Programme
2394:   \CommandIndex{makeindex} und/oder \CommandIndex{bibtex} erforderlich.
2395:
2396:   \DescOnlyNoIdx{cygwin\index{Probleme>cygwin}} Von der Erstellung
2397:   der \LaTeX"~Do\-ku\-men\-te zusammen mit dem Paket \fomsdt\ wird
2398:   unter \emph{cygwin}, einer Linux"~artigen\index{Linux} Umgebung
2399:   für Windowssysteme\index{Windows}, dringend abgeraten, da hierbei
2400:   gleich mehrere Probleme auftreten können.
2401:
2402:   Eines dieser Probleme ist, dass die Erstellung des Indexes zusammen
2403:   mit dem Stil \FileIndex{\fomidxist} fehlschlägt, da in der unter
2404:   \emph{cygwin} verfügbaren \LaTeX"~Umgebung einige Abhängigkeite
2405:   für die DIN"~gerechte, deutschsprachige Indexerstellung standardmäßig
2406:   nicht installiert sind!
2407:
2408:   \DescOnlyNoIdx{Seitenvorschub\index{Probleme>Seitenvorschub}}
2409:   Das Erzwingen eines Seitenvorschubes für einzelne Kapitel durch
2410:   Änderung des Kommandos \MacroIndex{\section} führt eventuell zu
2411:   dem unerwünschten Nebeneffekt, dass die Verzeichnisse der Stile
2412:   \FileIndex{\fomdsssty} und \FileIndex{\fomdessty} trotz angegebener
2413:   Option \OptionIndex{nopagebreak} nicht fortlaufend ausgegeben
2414:   werden. Dieses Problem kann durch den hier gezeigten Workaround
2415:   umgangen werden:
2416:
2417:   \begin{verbatim}
2418:    ...
2419:    \begin{document}
2420:    ...
2421:    \PrintContents
2422:    \PrintAbbrev
2423:    \PrintFigures
2424:    \PrintTables
2425:    ...
2426:    % Seitenvorschub zu Kapitelbeginn erzwingen!
2427:    \let\tempsection\section
2428:    \renewcommand\section{\clearpage\tempsection}
2429:    ... the writing ...
2430:    % Seitenvorschub zurücksetzen!
2431:    \let\section\tempsection
2432:    ...
2433:    \PrintListings
```

```
2434:    \PrintBibliography
2435:    \PrintGlossary
2436:    \PrintIndex
2437:    ...
2438:    \end{document}
2439:   \end{verbatim}
2440:
2441:   \DescOnlyNoIdx{\package{beamer}"~Klasse\index{Probleme>beamer}}
2442:   Wird \fomfod\ zusammen mit der Dokumentenklasse \DependsIndex{beamer}
2443:   geladen, dann treten hierbei verschiedentliche Konflikte und Fehler
2444:   in den Paketen \DependsIndex{caption} und \DependsIndex{subfig}
2445:   auf. Zwar sind diese Probleme den entsprechenden Paketentwicklern
2446:   bekannt, jedoch existiert derzeit noch keine geeignete Fehlerbehebung.
2447:
2448:   Aus diesem Grund wird das Laden des Pakets \package{subfig}
2449:   unterdrückt, wenn \fomfod\ zusammen mit der Klasse \package{beamer}
2450:   verwendet wird. Entsprechende Paketwarnungen werden während des
2451:   Kompiliervorgangs durch \fomfod\ ausgegeben.
2452:
2453:   Diese Unterdrückung hat leider auch zur Folge, dass eine Verwendung
2454:   des Makros \MacroIndex{\DoublePicture} nicht mehr möglich ist! Wird
2455:   \MacroIndex{\DoublePicture} unter diesen Umständen dennoch verwendet,
2456:   dann wird in der Präsentation anstelle der einzelnen Bilder eine
2457:   entsprechende Textmeldung angezeigt.
2458:
2459:   Wohlgemerkt, die zuvor beschriebenen Probleme treten nur dann auf,
2460:   wenn \fomfod\ zusammen mit der Dokumentenklasse \package{beamer}
2461:   verwendet wird!
2462:
2463:   \section*{\Depends}\addcontentsline{toc}{section}{\Depends}
2464:
2465:   Dieser Abschnitt verschafft einen Überblick über die Abhängigkeiten
2466:   des Paket \fomsdt\ zu anderen \LaTeX"~Pakete.
2467:
2468:   \DescDepends{a4wide} Das Paket zur Verwendung der gesamten verfügbaren
2469:   Seitengröße für DIN~A4 wird ohne Optionen durch \fomsdt\ geladen.
2470:
2471:   \DescDepends{caption} Das Paket zur Formatierung von Unter-- und
2472:   Überschriften für Abbildungen, Tabellen und Listings wird ohne
2473:   Optionen durch \fomfod\ geladen.
2474:
2475:   \DescDepends{color} Das Paket zur Unterstützung von farbigen Texten,
2476:   die in Dokumenten unter anderem für Hyperlinks verwendet werden, wird
2477:   ohne Optionen durch \fomsdt\ und \fomfod\ geladen.
2478:
2479:   \DescDepends{fancyhdr} Das Paket zur Unterstützung der Formatierung
2480:   von Kopf-- und Fußzeilen wird ohne Optionen durch \fomsdt\ geladen.
2481:
2482:   \DescDepends{fontenc} Das Paket zur Unterstützung verschiedener
2483:   Schriftarten wird mit der Option \option{T1} durch \fomsdt\ geladen.
2484:
```

```
2485:  \DescDepends{footmisc} Das Paket zur Unterstützung der Formatierung
2486:  von Fußnoten mit hängendem Einzug wird mit den Optionen \option{hang}
2487:  und \option{stable} durch \fomsdt\ geladen.
2488:
2489:  \DescDepends{gloss} Das Paket zur Erzeugung des Sachwortverzeichnisses
2490:  wird mit der Option \option{german} durch \fomdes\ geladen, wenn
2491:  dieser Stil zuvor mit der Option \option{glossary} geladen wurde.
2492:  Wurde die Option \option{glossrefpage} ebenfalls mit angegeben, dann
2493:  wird das Paket \package{gloss} zusätzlich mit der Option
2494:  \option{refpages} geladen.
2495:
2496:  \DescOnlyNoIdx{Achtung!\index{Glossar>Achtung}} Eventuell ist das
2497:  Paket \package{gloss} \emph{nicht} im \LaTeX"~System installiert und
2498:  muss deshalb nachinstalliert werden, wenn in der wissenschaftlichen
2499:  Arbeit ein Glossars verwendet werden soll! Empfohlen wird die
2500:  Installation des Pakets \package{gloss} in der Version v1.5.2 vom
2501:  26.~07.~2002 (oder höher).
2502:
2503:  \DescDepends{graphicx} Das Paket zur Grafikunterstützung wird ohne
2504:  Optionen durch \fomfod\ und \fomisc\ geladen.
2505:
2506:  \DescDepends{hyperref} Das Paket zur Unterstützung von Hyperlinks
2507:  wird mit den Optionen \option{colorlinks = true},
2508:  \option{bookmarks = true} und \option{plainpages = false} geladen.
2509:  Es wird durch \fomsdt\ nur dann geladen, wenn ein Dokument mit dem
2510:  Programm \CommandIndex{pdflatex} erstellt wird.
2511:
2512:  \DescDepends{ifpdf} Das Paket zur Prüfung des Ausgabeformats PDF
2513:  bei Verwendung des Programms \CommandIndex{pdflatex} wird ohne
2514:  Optionen durch \fomsdt, \fomdss\ und \fomdes\ geladen.
2515:
2516:  \DescDepends{ifthen} Das Paket zur Unterstützung von
2517:  Bedingungsabfragen wird zur intern Auswertung von Übergabeparameter
2518:  verwendet und durch \fomsdt, \fomfod, \fomdes\ und \fomisc\ geladen.
2519:
2520:  \DescDepends{inputenc} Das Paket zur Unterstützung des Zeichensatzes
2521:  für Eingabedaten wird mit der Option \option{latin1} durch \fomsdt\
2522:  geladen.
2523:
2524:  \DescDepends{listings} Das Paket zur Unterstützung von Codelistings
2525:  wird ohne Optionen durch \fomfod\ geladen.
2526:
2527:  \DescDepends{makeidx} Das Paket zur Unterstützung der Indexerstellung
2528:  wird ohne Optionen durch \fomdes\ geladen, wenn dieser Stil mit zuvor
2529:  der Option \option{index} geladen wurde.
2530:
2531:  \DescDepends{multicol} Das Paket zur Unterstützung der mehrerer
2532:  Spalten wird ohne Optionen durch \fomdes\ geladen, wenn dieser Stil
2533:  zuvor mit der Option \option{index} geladen wurde.
2534:
2535:  \DescDepends{natbib} Das Paket zur Erzeugung des
```

```
2536:   Literaturverzeichnisses wird mit den Optionen \option{square} und
2537:   \option{numbers} durch \fomdes\ geladen, wenn dieser Stil zuvor
2538:   \emph{nicht} mit der Option \option{nobibliography} geladen wurde.
2539:   In diesem Fall erfolgt die Erzeugung eines Literaturverzeichnisses
2540:   unter Verwendung des DIN"~gerechten Verzeichnisstils \defval{dinat}.
2541:
2542:   \DescOnlyNoIdx{Achtung!\index{Literaturverzeichnis>Achtung}}
2543:   Eventuell ist die Stildatei \FileIndex{dinat.bst} zur DIN"~gerechten
2544:   Literaturverzeichniserstellung \emph{nicht} im \LaTeX"~System
2545:   installiert und muss deshalb nachinstalliert werden!
2546:
2547:   \DescDepends{ngerman} Das Paket zur Unterstützung der
2548:   Besonderheiten der deutschen Sprache wird ohne Optionen durch
2549:   \fomsdt\ geladen.
2550:
2551:   \DescDepends{nomencl} Das Paket zur Erstellung des
2552:   Abkürzungsverzeichnisses wird nur dann geladen, wenn \fomdss\ zuvor
2553:   mit der Option \option{ab\-bre\-via\-tion} geladen wurde. In diesem
2554:   Fall wird \package{nomencl} mit der Option \option{german} geladen.
2555:   Wurde \fomdss\ ebenfalls mit der Option \option{abbrevrefpage}
2556:   geladen, dann wird \package{nomencl} zusätzlich mit der Option
2557:   \option{refpage} geladen.
2558:
2559:   \DescOnlyNoIdx{Achtung!\index{Abkürzungsverzeichnis>Achtung}}
2560:   \package{nomencl} ist grundsätzlich \emph{nicht} im \LaTeX"~System
2561:   installiert und muss deshalb eventuell nachinstalliert werden! Auf
2562:   einigen Systemen ist dieses Paket bereits in einer älteren und leider
2563:   auch inkompatiblen Version installiert. In diesem Fall sollte die
2564:   vorinstallierte Version des Pakets \package{nomencl} mindestens durch
2565:   die Version v4.2 vom 22.~09.~2005 (oder höher) ersetzt werden!
2566:
2567:   \DescDepends{rotating} Das Paket zur Unterstützung von rotierten
2568:   Texten wird ohne Optionen durch \fomsdt\ geladen.
2569:
2570:   \DescDepends{subfig} Das Paket Unterstützung von Untergrafiken wird
2571:   ohne Optionen durch \fomfod\ geladen.
2572:
2573:   \DescDepends{tabularx} Das Paket zur Unterstützung von Tabellen mit
2574:   fester Spaltenbreite wird ohne Optionen durch \fomfod\ geladen.
2575:
2576:   \DescDepends{textfit} Das Paket zur Skalierung von Texten wird ohne
2577:   Optionen durch \fomisc\ geladen.
2578:
2579:   \DescDepends{textpos} Das Paket zur Unterstützung spezieller Textboxen
2580:   wird mit der Option \option{overlay} durch \fomsdt\ geladen, wenn die
2581:   Klasse zuvor entweder mit der Option \option{moderntitle} oder der
2582:   Option \option{classictitle} geladen wurde.
2583:
2584:   \DescDepends{ulem} Das Paket zur Unterstützung von Unterstreichungen
2585:   im Abkürzungsverzeichnisses wird mit der Option \option{normalem}
2586:   geladen. Dieses Paket wird nur dann geladen, wenn \fomdss\ mit der
```

```
2587:   Option \option{abbreviation} geladen wurde. Weiterhin wird
2588:   \package{ulem} ohne weitere Einschränkung durch \fompar, ebenfalls
2589:   mit der Option \option{normalem}, geladen.
2590:
2591:   \StopEventually{%
2592:   \renewcommand\refname{Weiterführende Literatur}
2593:   \phantomsection\addcontentsline{toc}{section}{\refname}
2594:   \csname begin\endcsname{thebibliography}{1}
2595:   \bibitem{listings07}
2596:   \textsc{Brooks}, Moses; \textsc{Carsten}, Heinz: The \package{Listings}
2597:   Package. Januar 2007
2598:   \ifpdf
2599:   \href{http://tug.ctan.org/tex-archive/macros/latex/contrib/listings/listings.pdf}
2600:   {(Link zu: \tt{\small listings.pdf}).}
2601:   \else
2602:   URL http://tug.ctan.org/tex-archive/macros/latex/contrib/listings/listings.pdf.
2603:   \fi
2604:   \bibitem{tabularx99}
2605:   \textsc{Carlisle}, David: The tabularx package. Januar 1999
2606:   \ifpdf
2607:   \href{http://www.ctan.org/tex-archive/macros/latex/required/tools/tabularx.pdf}
2608:   {(Link zu: \tt{\small tabularx.pdf}).}
2609:   \else
2610:   URL http://www.ctan.org/tex-archive/macros/latex/required/tools/tabularx.pdf.
2611:   \fi
2612:   \bibitem{natbib07}
2613:   \textsc{Daly}, Patrick~W.: Natural Sciences Citations and References.
2614:   Februar 2007
2615:   \ifpdf
2616:   \href{http://tug.ctan.org/tex-archive/macros/latex/contrib/natbib/natbib.pdf}
2617:   {(Link zu: \tt{\small natbib.pdf}).}
2618:   \else
2619:   URL http://tug.ctan.org/tex-archive/macros/latex/contrib/natbib/natbib.pdf.
2620:   \fi
2621:   \bibitem{gloss02}
2622:   \textsc{Díiaz}, Jose Luis; \textsc{Bezos}, Javier: The \package{gloss}
2623:   Package. Juli 2002
2624:   \ifpdf
2625:   \href{http://tug.ctan.org/tex-archive/macros/latex/contrib/gloss/gloss.pdf}
2626:   {(Link zu: \tt{\small gloss.pdf}).}
2627:   \else
2628:   URL http://tug.ctan.org/tex-archive/macros/latex/contrib/gloss/gloss.pdf.
2629:   \fi
2630:   \bibitem{latex4wp03}
2631:   \textsc{Gonzato}, Guido: \LaTeX\ for Word Processor Users. Dezember
2632:   2003
2633:   \ifpdf
2634:   \href{ftp://ftp.ntg.nl/mirror/tex-archive/info/latex4wp/latex4wp.pdf}
2635:   {(Link zu: \tt{\small latex4wp.pdf}).}
2636:   \else
2637:   URL ftp://ftp.ntg.nl/mirror/tex-archive/info/latex4wp/latex4wp.pdf.
```

```
2638:   \fi
2639:   \bibitem{Goossens93}
2640:   \textsc{Goossens}, Michel; \textsc{Samarin}, Alexander;
2641:   \textsc{Mittelbach}, Frank: The \LaTeX\ companion. Addison-Wesley
2642:   Professional; Auflage: 2nd ed.; Mai 2004.
2643:   \bibitem{latex1}
2644:   \textsc{Jürgens}, Manuela: \LaTeX\ - eine Einführung und ein bisschen
2645:   mehr. März 2000
2646:   \ifpdf
2647:   \href{ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a026.pdf}
2648:   {(Link zu: \tt{\small a026.pdf}).}
2649:   \else
2650:   URL ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a026.pdf.
2651:   \fi
2652:   \bibitem{latex2}
2653:   \textsc{Jürgens}, Manuela: \LaTeX\ - Fortgeschrittene Anwendungen.
2654:   Oktober 1995
2655:   \ifpdf
2656:   \href{ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a027.pdf}
2657:   {(Link zu: \tt{\small a027.pdf}).}
2658:   \else
2659:   URL ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a027.pdf.
2660:   \fi
2661:   \bibitem{Kopka98}
2662:   \textsc{Kopka}, Helmut; \textsc{Daly}, Patrick~W.: A guide to \LaTeX.
2663:   Addison-Wesley Professional; 3 Sub edition; Januar 1999.
2664:   \bibitem{makeindex87}
2665:   \textsc{Lamport}, Leslie: \textsl{MakeIndex:} An Index Processor For
2666:   \LaTeX. Februar 1987
2667:   \ifpdf
2668:   \href{http://tug.ctan.org/tex-archive/indexing/makeindex/doc/makeindex.pdf}
2669:   {(Link zu: \tt{\small makeindex.pdf}).}
2670:   \else
2671:   URL http://tug.ctan.org/tex-archive/indexing/makeindex/doc/makeindex.pdf.
2672:   \fi
2673:   \bibitem{makeidx98}
2674:   \textsc{Mösgen}, Peter: Makeindex, Sachregister erstellen mit \LaTeX.
2675:   Mai 1998
2676:   \ifpdf
2677:   \href{http://www1.ku-eichstaett.de/urz/schriften/makeidx.pdf}
2678:   {(Link zu: \tt{\small makeidx.pdf}).}
2679:   \else
2680:   URL http://www1.ku-eichstaett.de/urz/schriften/makeidx.pdf.
2681:   \fi
2682:   \bibitem{lshort07}
2683:   \textsc{Oetiker}, Tobias; \textsc{Partl}, Hubert; \textsc{Schlegl},
2684:   Elisabeth; \textsc{Hyna}, Irene: The Not So Short Introduction to
2685:   \LaTeXe. Juni 2007
2686:   \ifpdf
2687:   \href{http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf}
2688:   {(Link zu: \tt{\small lshort.pdf}).}
```

```
2689:   \else
2690:   URL http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf.
2691:   \fi
2692:   \bibitem{epslatex06}
2693:   \textsc{Reckdahl}, Keith: Using Imported Graphics in \LaTeX\ and
2694:   pdf\LaTeX. Januar 2006
2695:   \ifpdf
2696:   \href{ftp://ftp.tex.ac.uk/tex-archive/info/epslatex.pdf}
2697:   {(Link zu: \tt{\small epslatex.pdf}).}
2698:   \else
2699:   URL ftp://ftp.tex.ac.uk/tex-archive/info/epslatex.pdf.
2700:   \fi
2701:   \bibitem{l2kurz03}
2702:   \textsc{Schmidt}, Walter; \textsc{Knappen}, Jörg; \textsc{Partl},
2703:   Hubert; \textsc{Hyna}, Irene: \LaTeXe-Kurzbeschreibung. April 2003
2704:   \ifpdf
2705:   \href{http://www.ctan.org/tex-archive/info/lshort/german/l2kurz.pdf}
2706:   {(Link zu: \tt{\small l2kurz.pdf}).}
2707:   \else
2708:   URL http://www.ctan.org/tex-archive/info/lshort/german/l2kurz.pdf.
2709:   \fi
2710:   \bibitem{nomencl05}
2711:   \textsc{Veytsman}, Boris; \textsc{Schandl}, Bernd: \package{nomencl}:
2712:   A Package to Create a Nomenclature. September 2005
2713:   \ifpdf
2714:   \href{http://tug.ctan.org/tex-archive/macros/latex/contrib/nomencl/nomencl.pdf}
2715:   {(Link zu: \tt{\small nomencl.pdf}).}
2716:   \else
2717:   URL http://tug.ctan.org/tex-archive/macros/latex/contrib/nomencl/nomencl.pdf.
2718:   \fi
2719:   \csname end\endcsname{thebibliography}
2720:
2721:   \phantomsection\addcontentsline{toc}{section}{Index}
2722:   \PrintIndex}
2723:
2724:   \end{document}
```

Glossar

Durchschuss Der Begriff *Durchschuss* stammt aus der Zeit, als literarische Werke noch von Hand und aus einzelnen in Blei gegossenen Buchstabenstempeln, den so genannten Lettern, gesetzt wurden. Dabei bezeichnet der Durchschuss vereinfacht ausgedrückt einen zusätzlichen, vertikalen Abstand zwischen zwei Textzeilen, also den Abstand, der sich genau zwischen der Stempelunterkante und der Stempeloberkante zweier untereinander liegender Lettern befindet. Zur Erzeugung eines zusätzlichen Abstandes legte der Schriftsetzer so genanntes Blindmaterial zwischen die einzelnen Buchstabenreihen.

In diesem Zusammenhang sollte auch darauf hingewiesen werden, dass der Durchschuss nicht mit dem Zeilenabstand zu verwechseln ist. Denn als Zeilenabstand wird die Distanz bezeichnet, die sich zwischen den Grundlinien zweier aufeinander folgender Textzeilen befindet, wohingegen der Durchschuss einen zusätzlichen Zwischenraum zwischen zwei Textzeilen bezeichnet [20, 22, 24]. (Seite 8)

Gleitobjekt Ein *Gleitobjekt* stellt ein Objekt dar, das im Fließtext eingefügt wird und zumeist der Darstellung eines bestimmten Zusammenhangs dient. Abbildungen und Tabellen sind gängige Beispiele für derartige Objekte. Sie werden deshalb als Gleitobjekt bezeichnet, weil ein Schriftsetzer sie im Fließtext beliebig verschieben kann, bis der Gesamteindruck des Textes „stimmig“ wirkt. Daher haben Gleitobjekte den großen Nachteil, dass ihre genaue Position innerhalb des Textes nicht vorherbestimmt werden kann! (Seite 17)

Hurenkind Die letzte Zeile eines Absatzes, welche zugleich die erste Zeile einer neuen Seite ist, wird im Druckerhandwerk allgemein mit dem Begriff *Hurenkind* bezeichnet. Derartig zerrissene Absätze gelten im Druckgewerbe als besonders schwere handwerkliche Fehler, da sie sich auf das gesamte Schriftbild und somit auf den Lesefluss besonders negativ auswirken [22, 24]. (Seite 12)

Java™ Hierbei handelt es sich um eine von der Firma SUN MICROSYSTEMS entwickelte objektorientierte Programmiersprache, die ursprünglich zur Ansteuerung von zum Beispiel Küchengeräten gedacht war. Java-Programme sind im Gegensatz zu Anwendungen in anderen Programmiersprachen dadurch gekennzeichnet, dass der einmal generiert Binärcode *ohne* erneute Kompilierung auf unterschiedlichen → *Plattformen* lauffähig ist. Dieses Verhalten wird unter anderem dadurch erreicht, dass zur Ausführung des jeweiligen Java-Codes ein plattformabhängiger Interpreter, eine so genannte Virtual Machine, verwendet wird. Dieser Umstand macht Java-Programme besonders interessant für jegliche Art von Internetanwendungen. (Seite 96)

Kapitälchen Hierbei handelt es sich um eine spezielle Form der Darstellung einzelner Buchstaben einer bestimmten Schriftart. Dabei werden alle Großbuch-

staben wie sonst auch dargestellt. Alle Kleinbuchstaben werden ebenfalls als Großbuchstaben, jedoch mit nur zirka Dreiviertel der Höhe der Großbuchstaben, dargestellt. (Seite 9)

Lesevorgang Während des Lesens führt das Auge ruckartige Sprünge – so genannte Sakkaden – durch, ohne hierbei die einzelnen Buchstaben vollständig wahrzunehmen. Dabei hält die Augenbewegung für den Bruchteil einer Sekunde inne. Diese als Fixationspunkte bezeichneten Stopps sind unwillkürlich und dienen der Erfassung der nächsten bis zu zehn Buchstaben in Leserichtung. Eine Verarbeitung des Gelesenen erfolgt innerhalb des Gehirns durch den Vergleich visueller Muster bereits bekannter Wörter [22, 10, 19]. (Seite 7)

Methodenüberladung Das Überladen von Methoden ist ein in der objektorientierten Programmierung gängiges Verfahren zur Bereitstellung gleichnamiger Funktionen, die über unterschiedliche Parameterlisten verfügen. Dieses Verfahren wird häufig dort eingesetzt, wo ein einziger Algorithmus auf unterschiedliche Datentypen angewendet werden kann. (Seite 32)

Office Ein Softwarepaket, das die elektronische Verarbeitung von Texten, die Erstellung tabellarischer Kalkulationen, die Erarbeitung von Präsentationen, die Manipulation von Graphiken und vieles mehr ermöglicht, wird im Allgemeinen unter dem Begriff *Office* zusammengefasst. Wie der englische Fachbegriff bereits vermuten lässt, kommen derartige Pakte zumeist in Büros zum Einsatz. Andererseits erfreuen sich auch viele private Anwender der reichhaltigen Möglichkeiten, die Office-Produkte in der Regel bieten. (Seite 3)

Patch Als *Patch* wird in der Sprache der Softwareentwickler die Korrektur eines Programmierfehlers bezeichnet. Häufig werden auch kleinere Softwareupdates unter diesem Begriff zusammengefasst. Dabei handelt es sich aber nur um eine saloppe Umschreibung für ein Paket zu Softwareaktualisierung. (Seite 82)

Plattform Im Umfeld der Informationstechnologie wird unter einer *Plattform* zuerst einmal nur die Hardware und im Speziellen sogar nur die Zentraleinheit eines bestimmten Computersystems verstanden. Im weiteren Sinne kann durch diesen Begriff aber auch das auf einem Computer installierte Betriebssystem gemeint sein. Diese etwas allgemeiner gefasste Definition des Begriffs *Plattform* wird beispielsweise immer dann verwendet, wenn Softwareproduzenten ihre Produkte als → *plattformenübergreifend* anpreisen. (Seite 95)

plattformenübergreifend In der Terminologie der Softwareentwicklung wird ein Quellcode genau dann als *plattformenübergreifend* bezeichnet, wenn dieser ohne (größere) Änderungen für unterschiedliche → *Plattformen* kompiliert werden kann. Bezogen auf bereits fertig kompilierte Anwendungen kann auch dann

von *plattformenübergreifender* Software gesprochen werden, wenn diese auf verschiedenen Betriebssystemen, zum Beispiel unter Windows® und unter Linux, ein identisches Verhalten aufweisen.

An dieser Stelle sollte aber auch festgestellt werden, dass beispielsweise die Firma MICROSOFT® unter einer *plattformenübergreifenden* Software exakt die Programme versteht, die auf den verschiedenen Windows®-Versionen lauffähig sind. Dabei können nicht einmal 16-Bit Programme unter der 64-Bit-Variante des Betriebssystems Windows® XP gestartet werden. (Seite 95)

Präambel Unter dem Begriff *Präambel* wird der Teil eines LaTeX-Dokuments verstanden, der sich zwischen den beiden Kommandos `\documentclass{}` und `\begin{document}` befindet. Hier werden in der Regel globale Einstellungen für das entsprechende LaTeX-Dokument hinterlegt. (Seite 31)

Schusterjunge Die erste Zeile eines Absatzes, welche zugleich die letzte Zeile der vorherigen Seite ist, wird im Druckerhandwerk allgemein mit dem Begriff *Schusterjunge* bezeichnet. Schusterjungen gelten gegenüber → *Hurenkindern* als weniger schwerwiegender Fehler [22, 24]. (Seite 12)

Textverarbeitung Unter einer *Textverarbeitung* wird allgemein ein Computerprogramm verstanden, dessen Aufgabe darin besteht, Texte durch einen Benutzer anlegen, öffnen, ändern, speichern und drucken zu lassen. Eine grundlegende Eigenschaft solcher Programme ist die Möglichkeit zur Texteingabe über die Tastatur des Computers. (Seite 3)

What you see is what you get (WYSIWYG) Ein Computerprogramm zur Erstellung von Textdokumenten, dessen besondere Eigenschaft darin besteht, den eingegebenen Text „druckfertig“ anzuzeigen, wird im Allgemeinen mit dem Begriff *WYSIWYG* beschrieben. Derartige Computerprogramme werden häufig auch als → *Textverarbeitung* beziehungsweise als Textverarbeitungsprogramm bezeichnet [26]. (Seite 4)

Literatur

[1] Beinert, Wolfgang: *Das Lexikon der westeuropäischen Typographie.* 2007. – URL http://www.typolexikon.de/s/satzbreite.html. – Zugriffsdatum: 2007-11-11

[2] Böttcher, Martin ; Institut für Informatik (Hrsg.): *Einführung in das wissenschaftliche Arbeiten.* Universität Leipzig. 2007. – URL http://wacko.informatik.uni-leipzig.de/de/Lehre/0506/SS/SemASKE/files?get=einfuehrung_in_das_wiss_arbeiten.pdf. – Zugriffsdatum: 2007-12-15

[3] Chen, Pehong ; Harrison, Michael A.: *Index Preparation and Processing.* Juli 1991. – URL http://www.ctan.org/get/indexing/makeindex/paper/ind.tex. – Zugriffsdatum: 2008-01-08

[4] Dorloff, F.-D.: *Hinweise zur Anfertigung wissenschaftlicher Arbeiten.* März 2006. – URL http://www.bli.uni-essen.de/lehre/Anfertigung.pdf. – Zugriffsdatum: 2007-11-19

[5] Energiewirtschaftliches Institut (Hrsg.): *Leitfaden zur Erstellung von Seminararbeiten.* Universität zu Köln. Juni 2007. – URL http://www.energie.uni-koeln.de/fileadmin/user/PDFs/Leitfaden_Seminararbeiten_Stand_0707.pdf. – Zugriffsdatum: 2007-12-02

[6] Fachbereich Wirtschaft (Hrsg.): *Richtlinien für die Erstellung einer schriftlichen Abschlussarbeit.* Hochschule Darmstadt. Juni 2006. – URL http://www.fbw.fh-darmstadt.de/Data/BA2D4935-235D-45DE-9C27-91F3E5CDE6CE/ccFiles/FF06FC81-4A95-4ADE-8B03-9ED729E0FA18.pdf. – Zugriffsdatum: 2007-12-15

[7] Fachbereich Wirtschaftswissenschaften (Hrsg.): *Formelle Anforderungen und Hinweise zur Bearbeitung von Diplomarbeiten/Seminararbeiten im Fach „Management kleiner und mittlerer Unternehmen" (PRO KMU).* Universität Siegen. 2007. – URL http://www.prokmu.de/lehre/pdf_allgemein/bearbeitungsrichtlinien_seminar_diplom.pdf. – Zugriffsdatum: 2007-12-15

[8] Harvard Business School (Hrsg.): *Citation Guide.* Oktober 2007. – URL http://www.library.hbs.edu/guides/citationguide.pdf. – Zugriffsdatum: 2007-11-19

[9] Hochuli, Jost: *Das Detail in der Typografie. Buchstabe, Buchstabenabstand, Wort, Wortabstand, Zeile, Zeilenabstand, Kolumne.* Neuauflage. Niggli Verlag AG, Zürich, November 2005. – ISBN 3-7212-0547-2

[10] Huestegge, Lynn ; Institut für Psychologie (Hrsg.): *Visuelles Abtasten während des Lesens.* Rheinisch-Westfälische Technische Hochschule Aachen. Mai 2007. – URL http://www.psych.rwth-aachen.de/ifp-zentral/upload/huestegge/Handout3a.pdf. – Zugriffsdatum: 2007-11-11

[11] Humanwissenschaftliche Fakultät (Hrsg.): *Hinweise zum Zitieren fremder Quellen in sozialwissenschaftlichen Arbeiten.* Universität zu Köln. 2007. – URL http://www.uni-koeln.de/ew-fak/seminar/sowi/politik/Hinweise%20zum%20Zitieren%20fremder%20Quellen%20in%20sozialwissenschaf%85.pdf. – Zugriffsdatum: 2007-11-16

[12] Institut für Betriebswirtschaftliche Geldwirtschaft (Hrsg.): *Empfehlungen für Studierende zur Anfertigung wissenschaftlicher Arbeiten.* Georg-August-Universität Göttingen. Oktober 2003. – URL http://ifbg.wiwi.uni-goettingen.de/downloads/allgemeines/Empfehlungen-zum-Anfertigen_Okt03.pdf. – Zugriffsdatum: 2007-12-08

[13] Kümpel, Thomas ; Seng, Anja: *Leitfaden zur formalen Gestaltung von Seminar- und Diplomarbeiten.* Juni 2007. – URL http://campus.bildungscentrum.de/nfcampus/UserFiles/File/formalesarbeiten.pdf. – Zugriffsdatum: 2007-12-02

[14] König, Anne Rose: *Alles Buch; Studien der Erlanger Buchwissenschaft VII.* 2004. – URL http://www.buchwiss.uni-erlangen.de/AllesBuch/Koenig/Koenig.pdf. – Zugriffsdatum: 2007-11-11

[15] Kowalski, Susann: *Leitfaden zur Erstellung von Seminararbeiten.* Juni 2007. – URL http://studieren.wi.fh-koeln.de/kurse/homepage/ftp/Anleitung%20Diplomarbeit.pdf. – Zugriffsdatum: 2007-12-02

[16] Krawinkel, Michael: *Leitfaden zur Anfertigung einer Studien-, Master-, Diplom- oder Doktorarbeit.* April 2006. – URL http://www.uni-giessen.de/fbr09/int-nutr/Diplom-und%20Doktorarbeiten%20im%20AK/Leitfaden%20Pruefungsarbeiten.pdf. – Zugriffsdatum: 2008-01-19

[17] Lamport, Leslie: *MakeIndex: An Index Processor For LaTeX.* Februar 1987. – URL ftp://tug.ctan.org/pub/tex-archive/indexing/makeindex/doc/makeindex.pdf. – Zugriffsdatum: 2008-02-14

[18] LJUBENOVA, Martina: *Auswirkung von Schriftart und Satz auf die Lesbarkeit und Lesegeschwindigkeit (Aufsatz im Studiengang: Medieninformatik).* 2004. – URL `http://www.medien.ifi.lmu.de/fileadmin/mimuc/mmi_ws0304/exercise/aufsaetze/Martina_Ljubenova.html`. – Zugriffsdatum: 2007-11-11

[19] PLETL, Heinz: *Aufsatz über den Leseprozess.* 2007. – URL `http://www.typoguide.de/downloads/leseprozess.pdf`. – Zugriffsdatum: 2007-11-11

[20] RENNER, Paul: *Die Kunst der Typographie.* 1. Auflage. MaroVerlag, Augsburg, August 2003. – ISBN 3-87512-414-6

[21] ROTERMUND, Hermann: *Gute Typographie auf Papier und im Web.* 2007. – URL `http://www.typoguide.de/downloads/aufsatzrotermund.pdf`. – Zugriffsdatum: 2007-11-11

[22] RUNK, Claudia: *Grundkurs Typografie und Layout.* 1. Auflage. Galileo Press GmbH, Bonn, Februar 2006. – ISBN 3-89842-406-5

[23] SCHOOR, Cornelia: *Richtlinien zur formalen Gestaltung von Hausarbeiten, Magister-, Bachelor- und Masterarbeiten.* 2005. – URL `http://www.tu-chemnitz.de/phil/elearning/files/Richtlinien_Manuskriptgestaltung.pdf`. – Zugriffsdatum: 2008-01-19

[24] SIEMONEIT, Manfred: *Typographisches Gestalten.* 3. verbesserte Auflage. Polygraph Verlag GmbH, Frankfurt am Main, 1989. – ISBN 3-87641-253-6

[25] TALBOT, Nicola: *makedtx v0.94b: a Perl script to help create a DTX file from source code.* August 2007. – URL `http://theoval.cmp.uea.ac.uk/~nlct/latex/packages/makedtx/makedtx.pdf`. – Zugriffsdatum: 2007-11-06

[26] TESCHNER, Helmut: *Druck- & Medien-Technik. Informationen, gestalten, produzieren, verarbeiten.* 11. Auflage. Fachschriften-Verlag, Fellbach, 2003. – ISBN 3-931436-88-8

[27] TEUBER, Mathias: *Kriterien für die Lesbarkeit von Texten.* 2007. – URL `http://www.medien.ifi.lmu.de/fileadmin/mimuc/mmi_ws0304/exercise/aufsaetze/Mathias_Teuber.html`. – Zugriffsdatum: 2007-11-11

[28] THE LATEX3 PROJECT (Hrsg.): *LATEX 2ε for class and package writers.* Februar 2006. – URL `http://www.dante.de/help/latex/guide/clsguide.pdf`. – Zugriffsdatum: 2007-11-06

[29] Wilhelm, Andreas: *Leitfaden zum Anfertigen wissenschaftlicher Hausarbeiten.* Mai 2006. – URL http://www.uni-kiel.de/sport/test/media/arbeitsbereiche/sportwissenschaft1/steckbriefe/wilhelm/Materialien/methoden/Leitfaden5_040506_Kiel.pdf. – Zugriffsdatum: 2008-01-19

[30] Winker, Gabriele ; Derboven, Wibke: *Einführung in wissenschaftliches Arbeiten.* 2007. – URL http://www.tu-harburg.de/agentec/lehre/ast/AST_Wissenschaft.pdf. – Zugriffsdatum: 2007-11-16

Index

A

B

C

D

E

F

G

M

N

O

P

Q

R

S

T

U

V

W

Z

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass die vorliegende Diplomarbeit von mir selbstständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere, dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Weiterhin erkläre ich, dass die Diplomarbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat.

Kaarst, den 17.02.2008

(Ort, Datum) (Eigenhändige Unterschrift)